# COMMODORE & AMIGA nr 11




NR INDEKSU 355216
ISSN 0867-8022

Cena 10 000 zł

listopad 1992 r.

MAGAZYN UŻYTKOWNIKÓW KOMPUTERÓW «COMMODORE»



DISKMASTERY

TEST: SKANER RĘCZNY ALFASCAN DO AMIGI









FLI — SUPERTECHNIKA DLA C-64

ŁAP WIRUSA — str. 7

*Jak zapewne zauważyliście, stosunkowo mało miejsca poświęcamy w C&A (przynajmniej do tej pory) licencjonowanym programom użytkowym na C-64 bądź na Amigę. Dzieje się tak z bardzo prostej przyczyny: programów takich, poza nielicznymi wyjątkami, zwyczajnie brak. To znaczy owszem, są, ale w 99% pochodzą z giełdy lub przegrywane są — co tu dużo mówić — nielegalnie, np. od kolegi. Programy tak zdobyte pozbawione są instrukcji, nie mają gwarancji, niektóre ich opcje mogą nie działać (biedny h a c k e r nie dał rady wszystkiego odszyfrować...), słowem nabytek jest co prawda tani, ale też połowiczny. Użytkownik nie ma jednak wyboru: albo kupi z giełdy, albo w ogóle, bowiem firmy zajmujące się sprzedażą oprogramowania komputerowego opierają swój byt prawie wyłącznie na wszelakiej maści grach.*

*Jest to błędne koło naszego rynku komputerowego. Dystrybutorzy software'u chcą jak najszybciej zarobić, więc nie zawracają sobie głowy „użytkami" — tylko gry zapewnią im stały dopływ gotówki. Poza tym zdają sobie sprawę, że nikt w Polsce nie kupi np. profesjonalnego edytora tekstów na Amigę, powiedzmy ProWrite 3.3, za dwa miliony złotych (tyle mniej więcej kosztowałby ten program po doliczeniu opłat celnych, podatku obrotowego itd.) skoro na giełdzie będzie on prędzej czy później dostępny za jedną dziesiątą tej ceny. Dla h a c k e r ó w i handlarzy tworzy się więc wymarzona sytuacja. Nie muszą obawiać się konkurencji, w świetle obecnego prawa nie są przestępcami, nie płacą żadnych podatków (stoją po prostu na ulicy, nie są nigdzie zarejestrowani) i nie ponoszą żadnej odpowiedzialności za niepoprawne działanie sprzedawanych programów. Zatem produkują oni coraz więcej pirackich kopii, natomiast poważne firmy wykazują z dnia na dzień coraz mniej zainteresowania sprowadzaniem (kupnem) autoryzowanych programów użytkowych.*

*Czy jest jakieś wyjście? Pierwsza odpowiedź nasuwa się od razu: niech parlament uchwali wreszcie już od dawna oczekiwaną ustawę antypiracką. Oczywiście, należy to uczynić jak najszybciej, ale sama ustawa to jeszcze nie wszystko. Na pewno zniknęłyby z giełd nielegalne kopie programów, ale pozostałoby przecież mocno zakorzenione przyzwyczajenie rodaków do przegrywania sobie cichcem, od kolegi, wszystkiego, co się da. Ponadto w początkowym okresie po wprowadzeniu nowego prawa przeciętny użytkownik zostałby w ogole pozbawiony możliwości zakupu oprogramowania, bo giełdy by „wyschły", a firmy w dalszym ciągu nie sprowadzałyby programów użytkowych z powodu ich zbyt wysokiej ceny.*

*Wiemy jednak, że krach na giełdzie (dosłownie) zbliża się nieuchronnie, a wraz z nim nastąpią miesiące (lata?) posuchy software'owej. Wobec całej grozy sytuacji postanowiliśmy (tj. redakcja C&A) włączyc się do — jak to się obecnie mówi — wspolnego budowania rzeczywistości i sztucznie wymusić zapotrzebowanie na programy użytkowe tak dla Amigi, jak i dla C-64. Jak to zrobimy? Będziemy po prostu opisywać ich coraz więcej, nie patrząc na to, czy pochodza z giełdy, czy od pana Kowalskiego, czy od renomowanej firmy. Mamy nadzieję, że skłonimy w ten sposób Czytelników, aby zaczęli nachodzić i molestować dystrybutorów na zasadzie: „No jak to, nie macie tego programu? Co z was za firma? Szkoda, mogliscie trochę zarobic, a tak cóż... idę na giełdę..." Wystarczy, żeby jedna duża albo kilka mniejszych firm „załapało" i zaczęło sprowadzać do Polski oryginalne wersje programów, wtedy inne pójdą za nimi albo... zbankrutują. A może zdarzyć się i tak, że jakiś obrotny biznesmen nakłoni zachodnich producentów do znacznego obniżenia ceny oferowanego oprogramowania (wszak dla kapitalisty lepiej sprzedać za 10 dolarów, niż nie sprzedać za 100 dolarów), wtedy można będzie z czystym sumieniem przestać obawiać się o przyszłość polskiego rynku software'owego.*

*Programy użytkowe „niewiadomego pochodzenia" zaczęliśmy już opisywać, że wymienię choćby Directory Opus v3.4 na Amigę, czy POLSCRIPT, FONTMASTER i TEXTOMAT na C-64. Publikujemy też kurs programowania w języku AMOS, którego gwarantujemy — w żadnym sklepie ani firmie się nie uświadczy.*

*W tym numerze zachęcamy do przeczytania artykułów o „Diskmasterach", programach antywirusowych (Amiga) i EXPERCIE — starym, ale oryginalnym module do C-64. Zamieszczamy także dwie recenzje legalnego software'u pochodzącego z firmy ALDERAN: Ortografia i Emulator Kickstartu 1.3 dla Amigi. Ponadto kontynuujemy cykle o AMOSIE, asemblerze 68000 i 6502. A dla zaawansowanych użytkowników C-64 nie lada gratka: prawie wszystko o technice FLI, czyli... ale to już sami przeczytacie. Zapraszamy!*

***CHRISTIAN GRZENKOWICZ***

# M·E·N·U

## AMIGA



## C-64



## ORAZ




magazyn użytkowników komputerów «COMMODORE»




*Redaktor naczelny:* KLAUDIUSZ DYBOWSKI
*Sekretarz redakcji:* CHRISTIAN GRZENKOWICZ
*Opracowanie graficzne:* JOLANTA PRZEŹDZIECKA
*Zdjęcia:* JERZY STOKOWSKI
*Aranżacja okładki:* DARIUSZ DUCKI

*Stali współpracownicy:* ANDRZEJ BOBEK (szef działu Amigi), RAFAŁ BORZYŃSKI, PIOTR CERKIEWNIK, BARTŁOMIEJ DRAMCZYK, JERZY DUDEK, MARIUSZ FERDYN, PAWEŁ GALAS, BARTŁOMIEJ KACHNIARZ, WOJCIECH KAZIMIERCZAK, ROBERT KULIŚ, PIOTR LISZEWSKI, RAFAŁ PIASEK, OLAF PRZYBYSZEWSKI, BARTOSZ SMAGA

*Redakcja:* ul. Wasilkowskiego 7, 02-776 WARSZAWA tel./BBS: 643-1840
*Kontakt z Czytelnikami:* pon-pt w godzinach 10.00-17.00
*Wydawca:* Spółdzielnia „Bajtek" ul. Wspólna 61 00-687 Warszawa tel./fax: 21-12-05
*Skład i druk:* Przedsiębiorstwo Wydawniczo-Poligraficzne „GRYF", Sp. Akc. Ciechanów
*Korekta:* KRYSTYNA WYDURSKA, MARIA GOŹDZIEWSKA
*Nr zlecenia:* 52302
*Nakład:* 72 tys. egzemplarzy

# z ostatniej chwili...

Gold Disk Inc. sprzedaje już swój najnowszy produkt Professional Draw 3.0, będący rozwinięciem wcześniejszej wersji. Jest to krok naprzód w ewolucji profesjonalnych programów graficznych, jak zapewniają autorzy (w tym sam prezes Gold Disk). Używając nowej wersji powinno pracować się szybciej i łatwiej. Skalowanie i rozmieszczanie obiektów pojedynczych jak i grupowych odbywa się prawie bez ingerencji użytkownika. To samo dotyczy oczywiście całej kolorystyki (kolor, czarno-biały, lub odcienie szarego). ProDraw 3.0 może w pełni współpracować z Professional Page 3.0, co daje wspaniałe możliwości edycji tekstu.

O O O

Best Amiga Tips And Secrets autorstwa Danny Atkina to książka, której tytuł mówi sam za siebie. Jest to najbardziej kompleksowe podejście do tematu-morza: Amiga. W opracowaniu tym autor zajmuje się problemami, cechami charakterystycznymi i wspólnymi dla całej rodziny Amig, od 500 do 3000. Raczkowanie, poznawanie budowy i zastosowanie danej maszynki, trudności i osobliwości wynikające z różnych konfiguracji sprzętowych, kłopoty z użytkowaniem oprogramowania oraz parafilozficzne rozważania natury ogólnoamigowej — to wszystko znajdziesz w tej książeczce. Dowiesz się także w jaki sposób transferować dokumenty i pliki tekstowe na PC i dlaczego ten gargantuicznych rozmiarów kalkulator jest gorszy od twojej amigusi. Jest tam również o ... jest, ale nie to jest najważniejsze. Najważniejsze jest przesłanie Danny Atkina: „Uwierz w swoją Amigę, poznaj jej możliwości i siłę a nie zaznasz znużenia." No i MOC będzie zawsze z tobą.

O O O

Firma Salectech opracowła, produkuje i zajmuje się dystrybucją AirMouse — nowej bezprzewodowej myszy. AirMouse Remote Control ma znaleźć szerokie zastosowanie w takich dziedzinach jak business, nauka i wszelkie formy szkolenia. To nowe narzędzie pracy dla wykładowców i szkoleniowców wszelkiej maści pozwoli wyeliminować większość wad operowania myszą tradycyjną.
AirMouse może współpracować z IBM i jego klonami (w tym w pełni obsługuje DOS czy Windows), Apple Macintosh, Commodore Amiga, Commodore CDTV oraz Philips CD-I.
AirMouse ma zasięg dziesięciu metrów. Jej jednostka nadawcza zasilana jest baterią 9 V i posiada lampki kontrolne wskazujące poziom naładowania baterii. Dopuszczalne odchylenie kąta kierunku wynosi około 30 stopni w pionie i około 60 stopni w poziomie.
Odbiornik, który jest zasilany z zasilacza 12 V posiada wyjście typu Serial (ośmio-pionowy Mini-Din typu żeńskiego), diody w kolorach czerwonym i zielonym sygnalizujące zdolność do pracy i odbierania impulsów z nadajnika AirMouse. AirMouse Remote Control w wyposażeniu firmowym posiada kable przyłączeniowe i kosztuje około sześciuset dolarów.

O O O

Nareszcie na rynku! Firma Designing Minds Software wychodząc naprzeciw oczekiwaniom biznesmenów i księgowych sympatyzujących z komputerami typu Amiga wypuściła na rynek swój nowy produkt nazwany Business Front, który jest jedny z nielicznych poważniejszych programów księgowo-rachunkowych na nasz najukochańszy Commodore-Amiga.
Business Front składa się z trzech głównych części, które w ogólnym zarysie można wyodrębnić jako edytor dokumentacyjno-rachunkowy, księgę kalkulacji i sprawozdawczości przychodów oraz księgę płatności i rozchodów.
Program charakteryzuje się pełną modułowością, pozwalającą na szybki dostęp do konkretnych danych bez rozwijania całego systemu. Pozwala na dowolne bilansowanie i przelewanie danych z roku na rok. Umieszcza do 32500 rekordów na jednym pliku oraz ma zabezpieczenie przed niepowołanymi szperaczami w postaci hasła.
Business Front można w pełni wykorzystać na Amidze 500, 2000 i 3000. Minimalny wymóg to 1 MB RAM, a maksymalny to oczywiście twardy dysk, który znacznie przyspieszy i ułatwi pracę. Cena programu oscyluje wokół stukilkudziesięciu dolarów.

O O O

MegaFortress to tytuł nowego symulatora lotniczego firmy Three-Sixty, który dotyczy oczywiście najnowszej odmiany Latającej Fortecy B-52H. Szkielet tej gry oparty został na powieści Dale Browna „Flight of the Old Dog", która cieszyła się ogromną poczytnością w Stanach Zjednoczonych. Megafortress to standardowo pomyślana gra z trzydziestoma misjami do wykonania i trzema teatrami działań operacyjnych. W każdej misji gracz może wcielić się w rolę pilota, nawigatora, specjalisty od elektronicznych systemów bojowych oraz oficera kierującego broniami ofensywnymi.
„Wielka, zła i uzbrojona po zęby" — taki mniej więcej jest wizerunek Latającej Fortecy zawarty w MegaFortress, w grze o trójwymiarowej grafice, w którą wszyscy entuzjaści tego typu symulacji powinni zagrać.
MegaFortress kosztuje około sześćdziesięciu dolarów, działa na Kickstarcie 1.3 i 2.04 ale potrzebuje przynajmniej 1 MB RAM.

O O O

Firma Lyra Group wprowadziła na rynek softwarowy swoje dwa nowe produkty: Tray i WIM 2.0.
Tray to nazwa programu, którego zastosowanie znacznie ułatwi i poprawi pracę ręcznych skanerów czterocalowych. Dzięki niemu nie będzie już problemu krzywości i rozmaitych odchyłek przenoszonych obrazów czy tekstu, co w praktyce niemal zrówna jakościowo skanery ręczne ze stacjonarnymi. Dla przeniesienia dokumentu formatu osiem i pół cala wystarczy tylko przejechać skanerem w osi pionowej wzdłuż obu połówek nie dotykając i nie przesuwając tegoż dokumentu a następnie, używając programu, dowolnie zestawić obie połówki i stworzyć pełną stronę. WIM to skrót od Windows Image Merge i jest to program, który w głównej mierze ma wspomagać Tray. Oczywiście, jak zapewniają producenci. WIM 2.0 może współpracować z wieloma programami i narzędziami służącymi kreowaniu obrazu. Jego główne zalety poza pomocą przy skanowaniu, przetwarzaniu i tasowaniu obrazu to na przykład możliwość zgrywania i zapisywania oryginalnych kolorów wykorzystujących technikę 24-bitową.

Oba przedstawione programy spisują się równie dobrze przy pracy z obrazem czarno-białym, w odcieniach szarości czy w kolorze, przy czym Tray kosztuje pięćdziesiąt dolarów a WIM 2.0 około osiemdziesięciu.

O O O

Jeżeli ktoś nie ma żadnych zdolności muzycznych, nie ma ochoty studiować instrukcji a tylko zabawić się, jeżeli lubi muzykę, chce jej posłuchać lub pofantazjować z nią, to powinien zakupić nowy produkt firmy Dr. T Music Software zatytułowany Boom Box. Jest to znacznie uproszczona, dużo swobodniejsza i zabawniejsza wersja istniejących programów muzycznych. Dzięki Boom Box można bardzo łatwo składać dowolne utwory korzystając z mnóstwa gotowych motywów, melodii i gadgetów. Naturalnie istnieje też możliwość pełnej pracy twórczej, przy wykorzystaniu zestawów instrumentów i dźwięków, wśród których nie braknie krzyków, płaczy, szlochów, porykiwań i oczywiście najbardziej oczekiwanej „pieśni" ciężkiej gitary.
Całość jest okraszona dużą porcją animacji i radosnego humoru. Tylko sześćdziesiąt dolców za nieskończoną ilość uśmiechu i zadowolenia.

O O O

Walt Disney Computer Softwre sprzedaje już Stunt Island w wersji na Amigę. Świat symulacji zyskał dzięki produktom Disneya nową jakość. Chodzi oczywiście o możliwość nagrywania i późniejszego odtwarzania wykonanych misji czy zadań. Teraz, dzięki grze Stunt Island mamy możliwość wykorzystania tego wszystkiego na skalę dotąd nie spotykaną.
Możemy latać, reżyserować i filmować, wybuchać, robić karkołomne sztuki itp.
To jednozdaniowe streszczenie nawet w jednej dziesiątej nie oddaje tego, na co pozwala Stunt Island. Do wyboru mamy czterdzieści różnych samolotów, od czasów pierwszej wojny do dnia dzisiejszego; ponad sześćset obiektów i kompleksów obiektów wliczając w to pojazdy, budowle wszelakiego typu, mosty i lotniska, które i tak można dowolnie łączyć w potrzebne zestawy.
Zabawa zaczyna się jednak dopiero po „nakręceniu" różnych scen. Teraz mamy możliwość miksowania, wycinania i podkładania ścieżki dźwiękowej porządnego kawałka 256-kolorowego materiału filmowego — słowem zasmakowania w prawdziwym montażu.

O O O

FractalPro 5.1 to już kolejny, po FractalPro 4.0, program firmy MegageM służący do tworzenia i obróbki 24-bitowych obrazów w wysokiej rozdzielczości. FPro 5.1 pozwala zdalnie sterować skryptami AmigaDOS i ARexx, umożliwia superszybkie tworzenie i odtwarzanie obrazów oraz posiada cztery nowe rodzaje obróbki wycinkowej.
FractalPro 5.1 wymaga koprocesora Fast Math i przynajmniej 2.5 megabajtów RAM, ale za to program został na tyle unowocześniony, że służyć może także użytkownikom 68881, 882 lub ultra-szybkiej 68040. W porównaniu z wersją 4.0 nowy FPro jest od czterech do trzydziestu pięciu razy szybszy i posiada tę zaletę, że nawet używając funkcji głębokiego *zoomu* nie tracimy czasu na długotrwałe przeliczanie, ponieważ wersja 5.1 używa 80-bitowego przetwornika IEEE. Program ten umożliwia równoczesne tworzenie obrazu 24-bitowego i korzystanie z 256-cio kolorowej palety barw zawartych w HAM. Jego zastosowanie pozwala na korzystanie z 16.384 różnych kolorów i całej masy odcieni czerwonego, zielonego i niebieskiego. Dodatkowo możemy zorganizować barwy w formacie 24-bitowym na ponad 100 sposobów, wliczając w to metodę „negatywną". FractalPro 5.1 pozwala na zapisanie obrazów w formie 24-bitowych IFF dla korzystania z nich przy pracy z innymi programami, takimi jak Imagine, Imagine-Master czy DCTV. Jeżeli w swojej Amidze mamy 24-bitową kartę graficzną Firecracker, to bez żadnego ustawiania parametrów możemy zastosować w połączeniu z nią FPro 5.1. Program ten automatycznie wyczuwa kartę i sprzęga się z nią pozwalając na korzystanie z rozdzielczości 768*482 i 1024*482, które są sterowane z Firecrackera. Dzięki specjalnemu interfejsowi FPro 5.1 może być sterowany i korygowany prostymi skryptamik używającymi ARexxa czy DOS-u. Dzięki temu nie tylko można polecić programowi automatyczne kreowanie długich sekwencji obrazów na potrzeby animacji, ale także uruchamiać je lub przesyłać do nagrania metodą video niemal „w locie".

Na podst. Genie
opr. Piotr Jarosiewicz

# SKANER RĘCZNY ALFASCAN DO AMIGI

Skaner jest jednym z podstawowych narzędzi potrzebnych przy wykorzystywaniu komputera (w naszym przypadku Amigi) do prac związanych z DTP czyli Desk Top Publishing. Podobnie zresztą jak samochód jest jednym z podstawowych środków komunikacji. Jednak istnieją różne samochody — osobowe, dostawcze, ciężarowe. Ponadto wśród osobowych mamy i naszą swojską syrenę 105, poloneza caro i mercedesa 600 SEL — są to samochody różnych klas. Nie inaczej jest ze skanerami...

Skaner ALFASCAN (dalej będę go nazywał AS) jest typowym skanerem ręcznym pracującym w kilku trybach i rozdzielczościach. Podobne urządzenie miałem przyjemność testować w jednym z Bajtków, więc siłą rzeczy będę bardziej opierał się na porównaniu z tamtym skanerem. Oba urządzenia są tej samej klasy, mało tego, mają prawie te same parametry — potrafią skanować w jednej z czterech możliwych rozdzielczości — 100, 200, 300 i 400 DPI (ang. dots per inch — punktów na cal). Jednak podobieństwo nie kończy się na tym. Oba urządzenia pozwalają na wybranie czterech trybów skanowania — od trybu „text" do „photo" z dwoma pośrednimi.

Tryb „photo" próbuje „wykropkować" skanowany obraz tak, aby można było uzyskać złudzenie odcieni szarości (jak na rysunku 1). Mimo, że nie jest to przyjemne dla oka, szczególnie gdy ogląda się taki skaning w powiększeniu, pozwala przedstawić choć namiastkę kolorów na wydruku (szczególnie na drukarce laserowej lub atramentowej).

Tryb „text" jest odwrotnością „photo" — tu obraz jest tworzony bezpośrednio. Elektronika skanera rozpoznaje, czy dany punkt ma być przedstawiony jako zapełniony czy nie, a granicą ustawia się specjalnym pokrętłem jasności znajdującym się z boku skanera. Efekty uzyskiwane w taki sposób możemy zobaczyć na rysunku 2. Skaning wykonany w trybie „text" nadaje się do późniejszej obróbki oprogramowaniem typu OCR (ang. optical character recognition — rozpoznawanie pisma).

Do pełnego opisu części skanującej brakuje jeszcze dwóch elementów — przycisku „Start", który należy trzymać wciśnięty w czasie skanowania, i diody LED powiadamiającej miganiem, że skaner przesuwa się zbyt szybko. Jeżeli dioda pali się światłem ciągłym, oznacza to, że skanowanie przebiega prawidłowo.

Dość jednak teorii, przejdźmy do praktyki.

## HARDWARE

Skaner AS dostajemy w bardzo gustownym pudełku, w którym znajduje się sam skaner z instrukcją obsługi, interfejs do Amigi (podłączany do gniazda równoległego — parallel), zasilacz, specjalna listwa (o której później) oraz koperta z oprogramowaniem i instrukcjami do niego — są to programy Scan&Save oraz MergeIt. Należy w tym miejscu dodać, że AS występuje w dwóch odmianach — standardowej oraz Plus. Różnica sprowadza się prawdopodobnie do innego oprogramowania, które dostaje kupujący — niestety nie jestem w stanie tego potwierdzić w stu procentach, gdyż dysponowaliśmy tylko wersją standardową, jednak z instrukcji można wyciągnąć takie wnioski. (Wersja pakietu NIE JEST wyszczególniona na pudełku.)

Instalacja nie jest specjalnie trudna — należy wyłączyć Amigę i podłączyć interfejs w odpowiednie gniazdko komputera. Następnie łączymy skaner z interfejsem, ale uwaga! Przy tej czynności nie należy się spieszyć ani używać siły, gdyż może to spowodować uszkodzenie wyjątkowo delikatnej wtyczki skanera. Ostatnią czynnością, którą należy wykonać, jest podłączenie zasilacza. Dalej pozostaje tylko zapoznanie się z...

## SOFTWARE

W zestawie, jak już napisałem, znajdują się dwa programy — Scan&Save oraz MergeIt. Na początek o tym drugim — jest to program pozwalający na łączenie niezależnych skaningów w jeden. Do czego jest to potrzebne: otóż dzięki temu można wygenerować pełny skaning ze strony formatu A4 — wystarczy połączyć dwa półprodukty programem MergeIt. Pomocna w tym będzie listwa znajdująca się w zestawie — pierwszy skaning wykonujemy opierając lewą krawędź skanera o listwę, a drugi — opierając prawą krawędź. Listwa jest tak zrobiona, że wykonane przy jej pomocy skaningi będą miały pewną część wspólną, co ułatwi ich połączenie w programie MergeIt.

Scan&Save (S&S) jest właściwym programem do skanowania. Gdy pierwszy raz wgrałem go (po uprzednim zainstalowaniu czcionek na twardym dysku — bez nich program „nie ruszy"), odniosłem wrażenie, że skądś go już znam. Otóż S&S jest po prostu bardzo okrojoną wersją programu TouchUp, który znajdował się w zestawie skanera testowanego przez mnie wcześniej.

S&S nie pozwala np. na obsługę formatów TIFF, PCX czy Mac, nie zawiera prawie żadnych narzędzi do retuszowania obrazu oraz nie posiada wielu innych opcji rzadziej używanych. Oznacza to, że za pomocą tego programu można właściwie tylko wykonać skaning. Na szczęście nie uszczuplono programu o możliwość zgrania obrazu w szarościach. W skrócie chodzi o to, że po wybraniu tej opcji S&S analizuje grupy punktów o wielkości 8x8 punktów i z nich tworzy kwadracik 2x2 punkty o odpowiednim odcieniu szarości. Podobnie było to robione w TouchUp-ie, jednak tam analizowana była grupa 6x6 punktów, a w wyniku TU generował tylko jeden punkt w jednym z 16-tu odcieni szarości.

W praktyce obrazy tworzone przez opisywaną opcję są bardzo „kwadratowe", widać duże *pixele* w rozdzielczości 320x256 i dopiero w 640x512 można uznać, że wygląda to „jako tako".

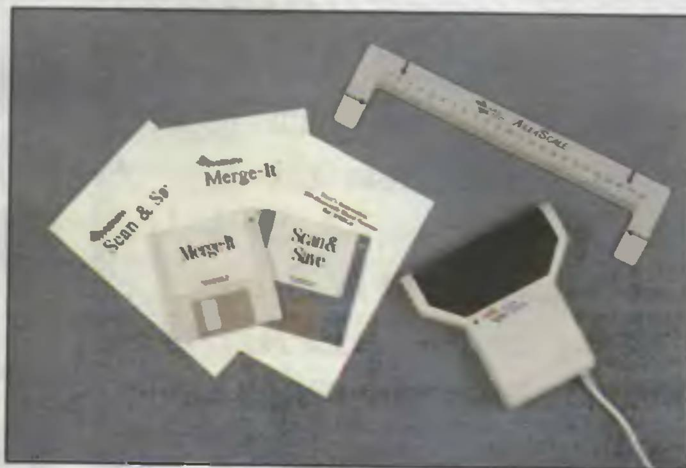
Jedną z opcji S&S, której brak w TouchUp-ie, jest możliwość zgrania obrazu z 256-ciu odcieni szarości. Jednak ponieważ na „standardowej" Amidze nie da się uzyskać takiej gradacji, zatem nie sposób odczytać taki obraz; opcję tę można śmiało zaliczyć do bezużytecznych „bajerów".

## WRAŻENIA

Najpierw o tym, co mi się nie podobało. Otóż na samym początku, zachęcony napisem „256-grayscale Hand Scanner for Amiga" umieszczonym na pudełku pomyślałem, że mam do czynienia z prawdziwym skanerem umożliwiającym operowanie 256-cioma odcieniami szarości. Niestety nie jest to prawdą — dopiero dołączone oprogramowanie pozwala na wygenerowanie obrazu w tylu odcieniach i to w dodatku nie do wyświetlenia na Amidze bez karty grafiki 24-bitowej.

Drugą rzeczą, która zwróciła moją uwagę jest jakiś błąd w programie powodujący, że gustowne ikony po lewej stronie ekranu w programie S&S są jak gdyby rozciągnięte a przez to zupełnie nieczytelne.

Podobało mi się natomiast to, że pro-




***Plastikowa listwa dostarczona w opakowaniu razem ze skanerem ułatwia (przynajmniej w teorii) wykonywanie skaningów z pełnego formatu A4***

gram ani raz się nie „powiesił", co często się zdarzało TouchUp-owi. Również pomysł z opisywaną listwą jest zacny, jednakże wymaga rozwinięcia, aby AS stanowił chociaż namiastkę skanerów stołowych formatu A4.

## PODSUMOWANIE

Według mnie zestaw ALFASCAN w wersji standardowej przeznaczony jest raczej wyłącznie do zastosowań nieprofesjonalnych (np. wydawanie gazetki szkolnej czy zabawa z obrazami). Wynika to nie tyle z niedoskonałości konstrukcji skanera, co raczej z powodu braku odpowiedniego oprogramowania wspomagającego użytkownika.

*RAFAŁ WIOSNA*

### ZALETY

+ możliwość zeskanowania pełnej strony A4,
+ teoretyczna możliwość uzyskania 256-ciu odcieni szarości,
+ prosta instalacja.

### WADY

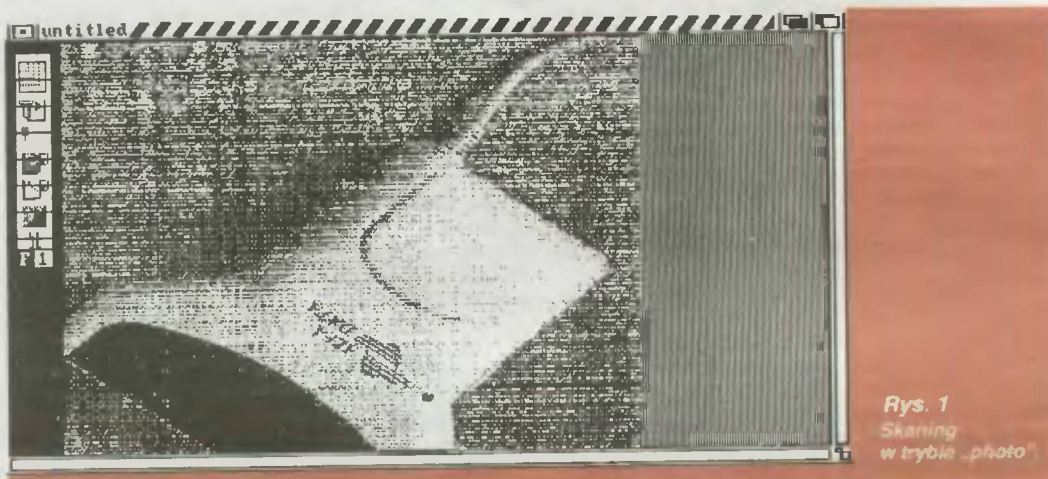
– nie najlepsze oprogramowanie,
– brak możliwości użycia formatów grafiki innych niż IFF,
– nie najlepsza konwersja kolorów na odcienie szarości.

### DANE TECHNICZNE

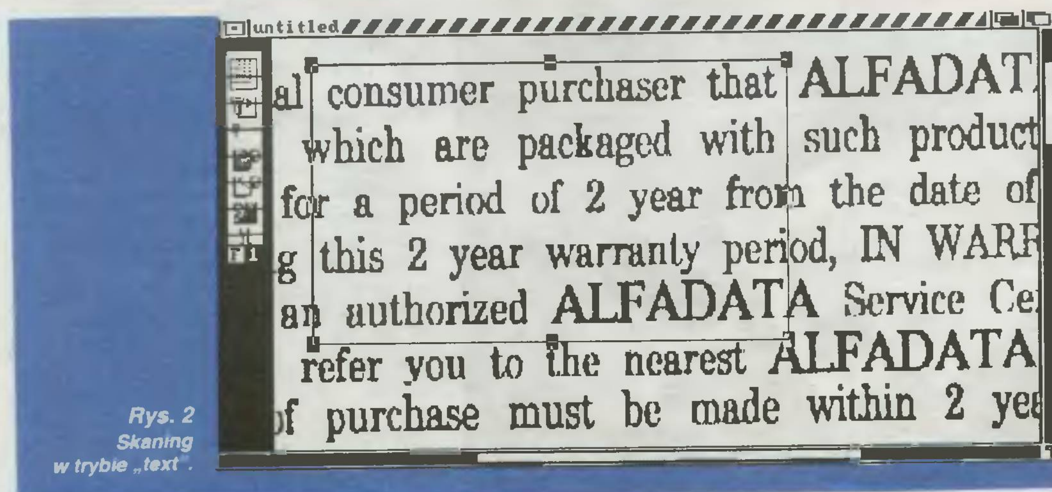
– współpracuje z Amigami: 500, 500+, 2000, 3000,
– współpracuje z systemem w wersji: 1.2, 1.3, 2.0,
– wymagana pojemność pamięci: min. 1 MB,
– rozdzielczość skanowania: 100, 200, 300, 400 DPI,
– tryby skanowania: text, 3x photo,
– szerokość skanowania: 105 mm,
– max. szybkość skanowania przy 100 DPI: 68 mm/s,
– max. szybkość skanowania przy 200 DPI: 34 mm/s,
– max. szybkość skanowania przy 300 DPI: 22 mm/s,
– max. szybkość skanowania przy 400 DPI: 17 mm/s,
– źródło światła skanowania: żółto-zielona dioda LED,
– pobór prądu 400 mA max. przy +12 V
– waga: 300 g
– wymiary 138x136x35 mm
– długość kabla: 1,8 m

**CENA:** ok. 2.850.000 zł (299 DM)

***DYSTRYBUTOR:***
***PROABIT, 05-500 Raszyn k/Warszawy, ul. Mickiewicza 14, tel. 56-08-91***




*Rys. 1*
*Skaning w trybie „photo"*




*Rys. 2*
*Skaning w trybie „text".*

# DELUXE PAINT IV — instrukcja obsługi

**Deluxe Paint IV jest zneny (przynajmniej tak by się wydawało) wszystkim użytkownikom Amigi. Jest to jeden z najlepszych programów graficznych, ale niewiele osób zne wszystkie jego możliwości. Większość Amigowcow bazuje na pirackich kopiach tego programu, a tym samym nie ma dostępu do instrukcji obsługi. Jeżeli się już taka znajdzie, jest ona marnej jakości odbitką (z nieczytelną czcionką i ilustracjami, które do niczego nie dadzą się przyrównać), najczęściej w języku angielskim lub niemieckim. Ja dostałem w swoje ręce instrukcję z prawdziwego zdarzenie.**

**Jest to tłumaczenie oryginalnej instrukcji do Deluxe Paint-a IV — otrzymujemy ją wraz z zakupem oryginalnej (nie pirackiej) wersji programu. Książka swoją jakością nie odbiega od zachodniego pierwowzoru, jedynie można mieć zastrzeżenia do ilustracji, które są mało ostre i niestety czarno-białe. Podręcznik został podzielony na kilka rozdziałow i jest on tak skonstruowany, że nawet „zielony" użytkownik (umiejący czytać) w bardzo krótkim czasie nauczy się posługiwać progremem.**

**Uczymy się od podstaw: posługiwania się myszką, okienkami, wybierania opcji z menu, rysowania linii prostych, okręgów, wypełniania zamkniętych obszarow, używania kolorow, wypełniania gradientami, używania szablonow itd. Poznajemy podstawy grafiki trojwymirowej: uczymy się perspektywy, skalowania obiektów, obracania ich w przestrzeni trojwymiarowej itd. Każda z opcji programu tłumaczonych w książce jest opatrzona odpowiednim przykładem i ilustracjami.**

**Najciekawszym rozdziałem, nawet dla osoby, ktora poznała już w pewnym stopniu program, będzie rozdział poświęcony animacji komputerowej. Można tam znaleźć wiele rad i sztuczek, które bez instrukcji są bardzo trudne do odkrycia. Dla początkującego natomiast, będzie to wspaniały wstęp do trudnej dziedziny jaką jest animacja komputerowa.**

**Poświęcono też jeden rozdział na zapoznanie użytkownika ze sposobem, w jaki Deluxe Paint IV operuje pamięcią komputera i w jaki sposób Amiga wyświetla na ekranie swoje tryby graficzne. Informacje te na pewno pomogą efektywnie wykorzystywać pamięć komputera, ktorej i tak zwykle nie mamy zbyt wiele.**

**Książka dołączana jest do każdej oryginalnej wersji programu Deluxe Paint IV, który rozpowszechniany jest przez firmę IPS Computer Group. Wydaje mi się, że dopiero podręcznik pozwala na dostrzeżenie wszystkich walorów Deluxe Paint-a IV i na pełne ich wykorzystanie.**

*BARTOSZ SMAGA*

**DELUXE PAINT — INSTRUKCJA OBSŁUGI, str. 290**
**Wydawca: IPS WYDAWNICTWO, Warszawa**





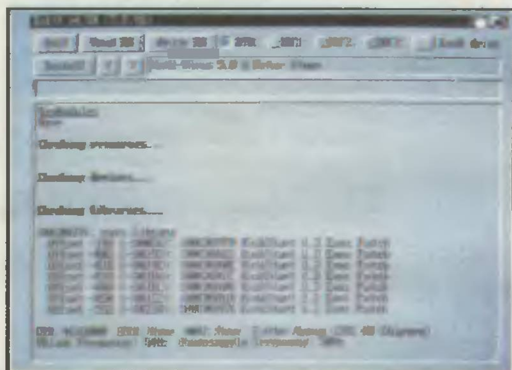
# ŁAP WIRUSA!

Zapewne każda osoba mająca jakikolwiek kontakt z komputerami zetknęła się z terminem „wirus komputerowy". Po wstępnym zaskoczeniu („Jak to, wirus w komputerze? Chyba mnie nie zarazi?") dowiadujemy się, że wirus komputerowy to nie żaden organizm, lecz krótki program mogący uszkodzić, a nawet zniszczyć owoce wielogodzinnej pracy. Najczęściej wirusy atakują dyskietki i dyski twarde podłączone do naszego komputera. Nie uszkadzają ich mechanicznie, lecz uszkadzają strukturę zapisu danych, co uniemożliwia ich późniejszy odczyt. Inne wirusy potrafią „zaczaić się" w pamięci i odczekawszy odpowiedni okres czasu skasować ją — jeżeli dostatecznie często nie zapisywaliśmy danych na dyskietkę, to utracimy je bezpowrotnie. Wirusy mogą także w inny sposób utrudniac pracę z komputerem, np. zamienić kierunki poruszania się myszki, pisac na ekranie głupie teksty czy odgrywać co jakiś czas swoją melodyjkę.

Na pewno pada w tej chwili pytanie — skąd biorą się wirusy? Wirusy pisane są przez ludzi znających na wskroś system operacyjny naszego komputera. Powstają jako „broń" przeciwko innym, nielubianym użytkownikom, powstają dla żartu, jako „odpady" z poważnej pracy, są także pokazem znajomości systemu. Rozprzestrzeniają się najczęściej poprzez sieci komputerowe, czyli telefonicznie, a także poprzez nośnik, czyli dyskietki.

Ze względu na specyfikę systemu operacyjnego Amigi, wirusy możemy podzielić na cztery rodzaje:

— wirusy *bootblockowe*, czyli mieszkancy pierwszych dwóch sektorów dyskietki. Są to wirusy zdecydowanie najpopularniejsze w Amigowym świecie i wykorzystują fakt, że po każdym resecie komputer startuje z dyskietki (bądź



*BootX V4.50*

dysku twardego) i pierwszą rzeczą jaką robi, to odczytuje właśnie *bootblock*.

— wirusy istniejące jako normalne pliki zapisane na dyskietce. Zmieniają one najczęściej sekwencję startową (startup-sequence) dyskietki w ten sposób, że wpisują tam swoją nazwę, co zmusza komputer przy jej wykonywaniu do uruchomienia wirusa. Są łatwe do wykrycia i do usunięcia.

— wirusy dołączające się do normalnych programów. Uruchomienie programu zarażonego takim wirusem uruchamia najpierw jego, a następnie dopiero sam program. Są to wirusy bardzo niebezpieczne ze względu na fakt, że są trudne do wykrycia.

— wirusy instalujące się w plikach systemowych, bądź je podmieniające. Są one bardzo niebezpieczne. Przykładowy wirus o nazwie Saddam instaluje się w programie o nazwie Disk Validator, znajdującym się na prawie każdej dyskietce, a czuwającym nad poprawnością jej zapisu. Inne wirusy tego typu potrafią zgnieżdżać się w bibliotekach systemowych, np. w icon.library.

Szczególnie narażeni są posiadacze twardych dysków, gdyż są one wymarzonym polem do popisu — wirusy mogą na nich tworzyć niewidoczne dla użytkownika partycje, instalować się w tablicy partycji itp. będąc jednocześnie odpornymi na usunięcie; niektóre potrafią nawet przetrwać formatowanie twardego dysku, czekając w pamięci komputera na zakończenie tej operacji.

Wraz z pojawieniem się wirusów powstały również programy mające na celu ich usuwanie. Programy te muszą być cały czas uaktualniane, gdyż pojawiają się ciągle nowe wirusy lub nowe wersje starego wirusa (np. istnieje ponad 10 wersji wirusa zwanego Lamer Exterminator). Istnieje niepisana zasada, że nie warto używać danej wersji programu antywirusowego dłużej niż 3—4 miesiące. Po tym okresie musimy postarać się o jego nowszą wersję, gdyż nasza może już nie wykrywać najnowszych odmian wirusów.

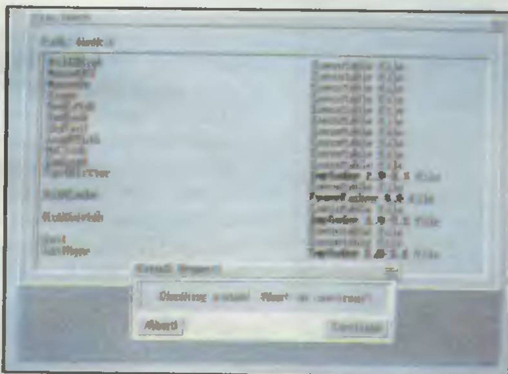
Po tym nieco przydługim wstępie mającym na celu zapoznanie z tematem mniej zorientowanych Czytelników, pragnę przedstawić kilka z najpopularniejszych programów antywirusowych dla Amigi.

## Virus Expert 2.0

Komercyjny program napisany przez Mac'a z grupy Katharsis, będący kontynuacją znanego programu rozprowadzanego jako Public Domain. Program ten ma spore możliwości, chyba jako jedyny potrafi przeanalizować kod nieznanego *bootblocku* i zadecydować, czy jest to wirus. Decyzjom tym nie należy całkowicie i bezgranicznie ufać, jednak sprawdzają się w około 80%. Czasami *bootblock* całkowicie bezpieczny oceniany jest jako wirus i odwrotnie. Pozwala na wykrycie wirusów instalujących się w pliku Disk Validator i na naprawę dyskietki uszkodzonej przez nie. Sprawdza także sekwencję startową dyskietki (jeśli taka istnieje).

## BootX 4.50

Jeden z najlepszych programów antywirusowych dla Amigi, rozprowadzany jako Freeware. Umożliwia wykrycie wszystkich spotykanych wirusów. BootX charakteryzuje



*VirusZ V2.19b*

się eleganckim wyglądem (a la Kickstart 2.0), korzysta z bardzo dobrej biblioteki reqtools.library. Jego obsługa jest dziecinnie łatwa. Po wgraniu program automatycznie sprawdzi pamięć naszego komputera, chroniąc nas przed możliwością obecnosci w niej wirusa. Następnie należy po kolei wkładać podejrzane dyskietki do dowolnej stacji. Program automatycznie sprawdzi *bootblock* i ewentualnie plik Disk Validator i w zależności od efektów wyświetli odpowiedni komunikat. Niestety nie potrafi on przy wkryciu nieznanego *bootblocku* przetestować go i ocenić (tak jak Virus Expert). Wynika z tego, że wszystkie podejrzane *bootblocki* należy bezwzględnie kasować (oczywiście dotyczy to tylko dyskietek zapisanych w formacie AmigaDOS). Program umożliwia również sprawdzenie wszystkich plików znajdujących się na dyskietce, bądź plików znajdujących się tylko w danym katalogu. W swej najnowszej wersji BootX przy napotkaniu na pliki spakowane rozpakowuje je i sprawdza. Niestety, na razie BootX rozpoznaje tylko format Power Packer-a i TNM Cruncher-a. Do innych opcji programu należy również możliwość tworzenia biblioteki nowych, nieznanych przez program *bootblocków*, reperacje dyskietek uszkodzonych przez Saddama, instalowanie w pamięci rezydentnego Link Virus Detector-a (istniejącego również jako oddzielny program) wykrywającego wirusy dołączające się do programów, zanim staną się aktywne. Mamy możliwość wyzerowania pięciu najważniejszych wektorów systemowych, najczęściej wykorzystywanych przez wirusy. Program posiada również bibliotekę kilkunastu bardzo dobrych *bootblocków* do dyspozycji dla użytkownika.
Dla początkujących BootX posiada wbudowany tzw. „help mode", po którego uaktywnieniu można uzyskać informację o każdej z funkcji programu. Program pozwala także na zmianę konfiguracji (kolory ekranu, rodzaj sampla alarmującego użytkownika itp.) i zgranie jej na dyskietkę.

## VirusZ 2.19b

Program mniej popularny niż BootX, lecz charakteryzujący się wcale nie mniejszymi możliwościami. Korzysta także z biblioteki reqtools.library lecz w przeciwieństwie do programów wyżej przedstawionych nie otwiera własnego ekranu lecz jedynie małe okno na ekranie Workbench-a. Daje nam możliwość sprawdzenia *bootblocku* dowolnej dyskietki, pliku Disk Validator a także wszystkich plików na dyskietce. Dzięki bibliotece decrunch.library (z której korzysta) przy sprawdzaniu plików rozpoznaje format ok. 30 *cruncherów*, które rozpakowuje i sprawdza. Ubocznym zastosowaniem programu może więc być sprawdzanie, jakim *cruncherem* został spakowany dany plik. Tak samo jak programy wyżej wymienione pozwala na naprawę dyskietki uszkodzonej przez Saddama.

## Virus Checker 6.15

Program bardzo podobny do wyżej wymienionego, tak jak i on nie otwiera własnego ekranu, a jedynie małe okno na ekranie Workbench-a. Jest bardzo popularny, jednak możliwościami ustępuje programowi VirusZ. Pozwala na wykrycie wirusów *bootblockowych*, file-owych i w Disk Validatorze, sprawdza pamięć, reperuje dyskietki uszkodzone przez Saddama.

Wszystkie wyżej wymienione programy (poza Virus Checkerem) pozwalają na nagranie dowolnego *bootblocku* na dyskietkę w postaci pliku i dokonanie czynności odwrotnej. Programy BootX i VirusZ zostały zamieszczone na naszej dyskietce Public Domain nr 2. Osobom nie mającym tych programów w swoich zbiorach radzę ją zamówić, a następnie sprawdzić swoje dyskietki. Efekt może być czasami interesujący...

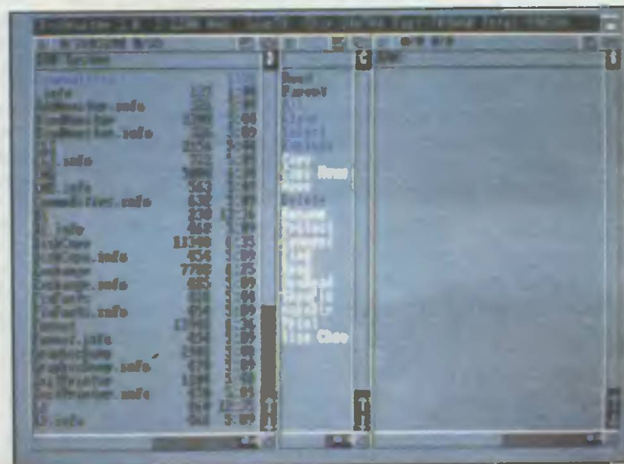
*HIGHTOWER*



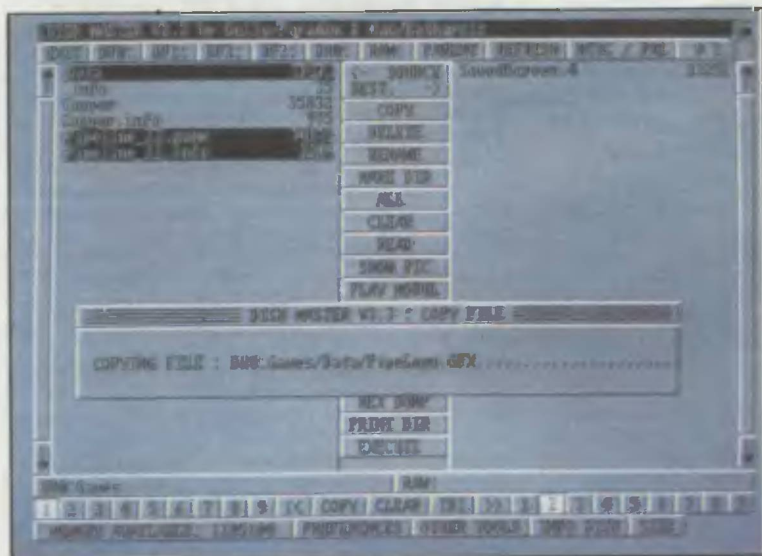
# DISKMASTERY

System operacyjny Amigi, mimo wielu zalet i przyjazności dla użytkownika, przy pracy z poszczególnymi plikami odkrywa swoje ciemne strony. Przeniesienie danego pliku z dyskietki na dyskietkę z poziomu Workbench-a jest jeszcze w miarę łatwe, wystarczy przenieść jego ikonę. Lecz co zrobić, gdy program nie ma ikony, nie mówiąc już o stworzeniu nowego katalogu dla niego. Żmudne operowanie komendami CLI może doprowadzić do szału mniej cierpliwego użytkownika. W celu ułatwienia pracy przy takich operacjach, powstały programy mające na celu czynienie tego w sposób łatwy i przyjemny. Programy te nazywane są ogólnie kopierami, bądź od nazwy najpopularniejszego z nich — „diskmasterami".

Od początku istnienia Amigi powstało bardzo dużo programów tego typu, tak, że początkującemuu użytkownikowi trudno wybrać ten najlepszy i najbardziej sprzyjający jego potrzebom i konfiguracji sprzętowej. Chciałbym krótko przedstawić te z nich, które uważam za godne uwagi.



*Disk Master V2.0*



*Disk Master V3.3*

## DISK MASTER 1.4

Jeden z najpopularniejszych programów tego typu, może już nie najmłodszy (1987). Był on jednym z pierwszych programów nadających się naprawdę do pracy i oferujących spore możliwości. Posiada on wszystkie podstawowe operacje tj. COPY, MOVE, DELETE, RENAME itd. a także wiele innych jak SHOWPIC, PLAYSND. Oferuje on również możliwość formatowania i kopiowania dyskietek, a także ułatwia korzystanie z programów archiwizujących tj. Zoo, Arc i LhArc. Do zalet tego programu należy zaliczyć możliwość dodawania dziesięciu własnych komend (tworzonych przez użytkownika). Możemy zainstalować pod nim takie programy jak PPMore czy PPShow, które pozwalają nam na współpracę z plikami spakowanymi Power Packerem. Niestety jest on już programem nieco przestarzałym i nie wytrzymuje konkurencji programów najnowszych.

## DISK MASTER 3.3

Nie jest to kontynuacja poprzednich programów, ale program całkiem innego autora (Gully/Paradox V3.0) i poprawiony przez Mac'a z Katharsis. Napisany całkowicie w asemblerze (poprzednie — w języku C), co wpływa na jego długość — program po spakowaniu zajmuje na dyskietce tylko 24 kilobajty (!). Bynajmniej nie oznacza to, że charakteryzuje się małymi możliwościami, wręcz przeciwnie. Poza standardowymi funkcjami oferowanymi przez inne programy tego typu, Disk Master 3.3 ma całkiem odrębny ekran funkcji dodatkowych.

Oto co oferuje: formatowanie dysków (z weryfikacją lub bez), tzw. *quick format* czyli formatowanie jedynie ścieżek 0 i 40, sprawdzanie poprawności zapisu danych na dysku, *virus killer* umożliwiający obejrzenie zawartości *bootblocku* i ewentualnej jego zmiany, *block editor*, czyli moduł dający nam możliwość obejrzenia całej dyskietki heksadecymalnie lub w kodach ASCII (także jedno i drugie naraz), kopiowanie dysków. Program ma również możliwość przełączania przez użytkownika swego głównego ekranu z trybu PAL na NTSC i odwrotnie. Bardzo dobrą poprawką dokonaną przez Mac'a było dodanie mu procedur, dzięki którym możemy oglądać obrazki, słuchać modułów i czytać teksty spakowane Power Packerem. Program ma jeszcze kilka błędów, lecz już teraz jest bardzo użyteczny i przydatny.

## DISK MASTER 2.0

Program tego samego autora co poporzednio omówiony, ma jednak dużo większe możliwości. Wykorzystuje okna systemowe otwierane na własnym ekranie lub na ekranie Workbench-a. Standardowo otwierane są dwa okna przeznaczone na katalogi i jedno z komendami, lecz mamy możliwość otwarcia dowolnej ich ilości. Także usytuowanie tych okien zależy jedynie od użytkownika. Program pozwala na dołączenie dowolnej ilości własnych komend w oknie komend lub w rozwijanych menu. Plik konfiguracyjny programu jest normalnym skryptem, czyli ciągiem występujących po sobie komend charakterystycznych dla Disk Master-a. Niestety, bez instrukcji nie jesteśmy w stanie wykorzystać wszystkich możliwości programu ze względu na bogactwo tych poleceń. Program pozwala na współpracę z dowolną ilością programów archiwizujących, jednak nie w sposób tak wygodny jak chociażby Directory Opus.

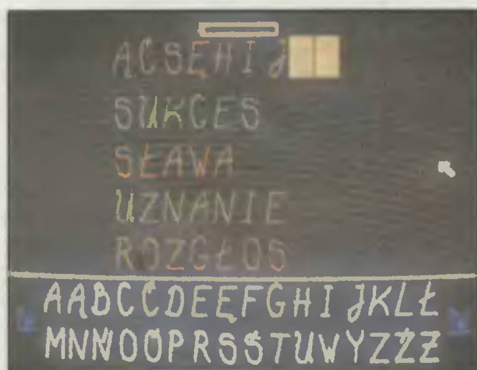
## FILE MASTER 2.1

Program zdobywający ostatnio coraz większą popularność i nie bez racji. Jest on napisany w asemblerze, co decyduje o jego szybkości. Oprócz komend występujących też w poprzednich programach ma funkcje dodatkowe, m.in. ma wbudowany całkiem niezły i szybki edytor plików, co jest raczej rzadkością w programach tego typu, a także bardzo wygodny edytor dyskietki dający możliwości znacznie większe niż Disk Master 3.3. Do innych, rzadziej spotykanych funkcji można zaliczyć opcję MODINFO, która podczas odgrywania dowolnego modułu z *trackerów* podaje nam informację o jego autorze, długości, a także ilości, nazwach i długości sampli.

Inne ciekawe opcje to: CRUNCH i DECRUNCH — czyli pakowanie i rozpakowywanie plików w standardzie Power Packera (najpopularniejszych), JOIN — łączenie dwóch plików w je-








*„Skojarzenia" — jedna z czterech gier zawartych w programie*

# ORTOGRAFIA — GRA SŁÓW

Ostatnio miałem przyjemność testować kolejny, całkowicie polski program edukacyjny dla Amigi o nazwie „Ortografia — Gra Słów". Właściwie jest to zestaw czterech gier edukacyjnych rozwijających wyobraźnię, refleks oraz powiększających zasób słów.

Pierwsza z gier pt. „Krzyżówka" polega na odnajdywaniu w tablicy liter wszelkich wyrazów o długości od trzech do dziesięciu liter. Punktacja uzależniona jest od długości wskazywanych słów.

Celem gry pt. „Skojarzenia" jest odgadnięcie wyrazu, którego długość znamy. Komputer co pewien czas podaje nam wyrazy naprowadzające.

Z kolei „Anagramy" polegają na odgadnięciu wyrazu, którego długość oraz wszystkie litery występujące w wyrazie są znane. Litery ułożone są w przypadkowej kolejności. Należy ułożyć je we właściwej kolejności, tak, aby utworzyły wyraz.

I ostatnia z gier, „Wisielec", znana jest na pewno wszystkim ze szkoły. Polega na odgadnięciu pewnego wyrazu, o którym wiadomo tylko, jaką ma długość. Wskazujemy myszką litery, które — jeżeli istnieją w wyrazie — pojawiają się na właściwych miejscach.

Program stanowi bardzo dobrą rozrywkę i może wciągnąć użytkownika na długie godziny. Ma wbudowany słownik ortograficzny zawierający 10000 słów. Walory edukacyjne programu są wysokie np. dzięki temu, że błędnie napisane słowo nie pojawia się na ekranie i uczeń nie zapamiętuje wzrokowo źle napisanego wyrazu. Ukazują się (i utrwalają w pamięci) tylko wyrazy napisane prawidłowo. Program, chociaż przeznaczony jest dla uczniów szkół podstawowych, może być świetną zabawą również dla dorosłych. Jego obsługa jest bardzo prosta, wszystkie funkcje wybiera się myszką.

W pudełku oprócz dyskietki znajduje się także 4-stronicowa szczegółowa instrukcja nie pozostawiająca żadnych wątpliwości co do obsługi programu.

„Ortografia" działa na każdej Amidze w każdej konfiguracji. Nie zauważyłem żadnych nieprawidłowości w funkcjonowaniu programu. Jedynym niedociągnięciem jest brak jakichkolwiek efektów dźwiękowych i moim zdaniem nieco zbyt prosta grafika.

Myślę, że „Ortografia" jest wprost nieocenioną pomocą dla osób mających kłopoty z poprawną pisownią. Poprzez zabawę, zgodnie z zasadami dydaktyki języka polskiego, można nauczyć się naprawdę sporo. Należy cieszyć się, że rodzimi programiści tworzą tego typu oprogramowanie — przecież nikt za Polaków nie zrobi takiego programu. Za przystępną cenę otrzymujemy wspaniały pakiet czterech gier edukacyjnych, spełniających doskonale postawione przed nim zadanie.

***JERZY DUDEK***

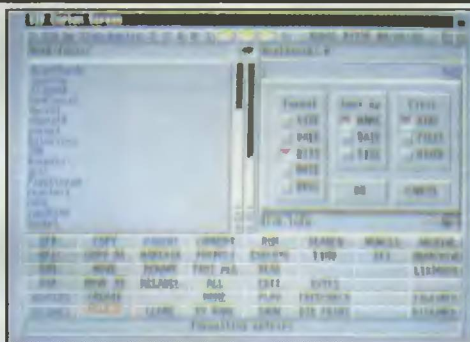
*Plansza tytułowa*




**DYSTRYBUTOR:**
**ALDERAN S.C., ul. Korotyńskiego 19a/55, 02-123 Warszawa, tel. 659-18-21**



*Filemaster V2.1*



*SID V2.0*

den, CUT — możliwość wycięcia z danego pliku dowolnego jego fragmentu. File Master pozwala również na dodawanie własnych opcji i komend. Posiadana przeze mnie wersja tego programu jest jedynie wersją demonstracyjną i ma zablokowaną możliwość zgrywania własnej konfiguracji na dyskietkę. Pełną wersję można otrzymać po rejestracji. Warto zaznaczyć, że mimo ogromnych możliwości programu (są to właściwie trzy programy w jednym) ma on tylko 80 KB długości!

## SID 2.0

Niektórzy z Szanownych Czytelników zapewnie znają jego starsze wersje (np. 1.06), pragnę ich zapewnić, że jest to jedynie namiastka najnowszej wersji. Sid posiada ogromne możliwości ustawiania własnej konfiguracji a także wyglądu programu przerastające momentami program Directory Opus, do którego można go porównać. Niestety Sid może być użyteczny właściwie tylko dla użytkowników twardego dysku lub co najmniej dwóch stacji dysków.

W swojej podstawowej wersji Sid umożliwia jedynie kopiowanie i normalną edycję plików (zmiana nazwy, kasowanie itd.). Spowodowane jest to tym, że nie ma on wbudowanych żadnych programów umożliwiających przeczytanie danego tekstu, obejrzenie obrazka, wysłuchanie dowolnego sampla czy modułu muzycznego itp., i w celu dokonania tych czynności musi on załadować odpowiedni program z dysku. Dla jednych może być to wada, dla innych ogromna zaleta, gdyż daje nam to możliwość używania własnych, ulubionych programów służących temu celowi.

Jednocześnie rozpowszechniane są dwie wersje SID-a. Pierwsza, tzw. „Professional", ma wbudowany program dający możliwość zmiany w konfiguracji i nagranie jej na dyskietkę; druga, tzw. „Personal" jest o 100 KB krótsza od poprzedniej (jej długość to ok. 150 KB), nie daje jednak możliwości zmiany w konfiguracji a przeznaczona jest dla użytkowników nie dysponujących nadmiarem pamięci RAM.

## DIRECTORY OPUS 3.41

Zdecydowanie najlepszy program tego typu na Amigę będący w stanie zaspokoić potrzeby najbardziej wymagającego użytkownika. Poświęcony mu był osobny artykuł w C&A nr 8/92.

***HIGHTOWER***

# EMULATOR 1.3

Gdy jakiś czas temu na rynku pojawiła się Amiga 500+, za jeden z jej największych mankamentów uważano niepełną kompatybilność z Amigą 500. Wynikało to z faktu, że w nowej Amidze zastosowano nowy system operacyjny, a mianowicie Kickstart 2.04. Przez to znaczny procent gier, wszelakiej maści dem, jak również nieco programów użytkowych nie działało. Tylko małą część stanowiły te „padające" z powodu zastosowania w nowej Amidze kostek specjalizowanych z Amigi 3000 — głównym winowajcą był sysem operacyjny. W takiej sytuacji posiadacze Amigi 500+ mieli tylko jedno, do niedawna, wyjście: uszczuplić swój portfel o ok. 600 tys. złotych montując w swojej Amidze dodatkowy Kickstart i przełącznik do wyboru ROM-u. Z chwilą jednak pojawienia się programu będącego przedmiotem tego opisu, posiadacze Amigi 500+ otrzymali alternatywne, programowe rozwiązanie trapiącego ich problemu.

## ZASTOSOWANIE I URUCHOMIENIE

Jedynym bowiem zadaniem Emulatora 1.3 jest „podmienienie" systemu operacyjnego Amigi na stary, dobry Kickstart 1.3, tak, by programy stosujące „nielegalne" praktyki zaczęły działać. Podmiana systemu odbywa się na dość prostej, zdawałoby się, zasadzie: system w wersji 1.3 wczytywany jest do pamięci RAM Amigi 500+, po czym wszystkie wektory systemowe zostają tak ustawione, że cała kontrola przechodzi „w ręce" systemu w wersji 1.3. W praktyce sprawa jest nieco bardziej skomplikowana — wspomnę tu choćby o fakcie, że oryginalny system operacyjny dostosowany jest do pracy tylko i wyłącznie w obszarze ROM-u (czyli od $FC0000 do $FFFFFF), teraz zaś musi działać w zupełnie innym obszarze adresowym. Tak więc system musiał zostać relokowany, czyli mówiąc po polsku, tak przerobiony, by mógł funkcjonować pod innymi adresami. Poza tym przejęcie kontroli przez nowy system oznacza, że musi on zacząć kontrolować wszystkie, bardzo skomplikowane struktury (jak choćby tablice zadań, serwerów przerwań, device-ów i stu innych rzeczy). W praktyce niemożliwe byłoby chyba dokonanie tego bez uciekania się do niezwykle prostego triku: podmieniane jest tylko tyle wektorów, by system przejął kontrolę po zresetowaniu Amigi — po resecie system, już w wersji 1.3, sam tworzy sobie wszystkie struktury. Jedyną strukturą, która jest fałszowana, jest mapa pamięci RAM — należy bowiem wyłączyć z użytku 256 KB zajmowane przez system w wersji 1.3.

Przejdźmy jednak do praktyki. Aby uruchomić Emulator 1.3 należy zresetować Amigę, po czym włożyć dyskietkę z programem do wewnętrznej stacji. Od tej pory wszystko odbywa się już samo. System odczytuje i uruchamia *bootblock* i na tym jego rola się praktycznie kończy. Po chwili kontrolę przejmuje program zapisany na dyskietce — na ekranie pojawia się plansza z informacjami o autorze i firmie Alderan, jednocześnie w tym samym czasie nowy system jest ładowany do pamięci. Pojawiające się, w miarę ładowania, na ekranie prostokąty informują nas o tym, ile danych już jest w pamięci. Gdy transmisja zostanie ukończona, program prosi o wyjęcie dyskietki — i to wszystko, w tym momencie automatycznie następuje reset, po chwili zaś na ekranie gości słynna „żebrząca rączka". W tej chwili mamy do dyspozycji Amigę 500+ z 768 KB CHIP-RAM i systemem w wersji 1.3.

## PRAKTYKA

Program sprawdzałem na dość rozbudowanej Amidze 500+ — twardy dysk, 9 MB RAM, trzy stacje dysków, AT-Once itd. W pewnej fazie zacząłem (oczywiście po kolei) odłączać przystawki, aby sprawdzić, czy program nie jest przypadkiem zbyt kapryśny i wymagający. Wyniki były naprawdę imponujące: Emulator 1.3 ani razu nie zawiesił się, nie było też najmniejszych kłopotów z samymi przystawkami. Jedyny przypadek, kiedy program „padł" miał miejsce po wymianie portu Amigi na uszkodzony. Hardware Amigi nie był wtedy w stanie poprawnie oczytać zabezpieczenia przed kopiowaniem Emulatora 1.3.

Tak więc pierwszym etapem było sprawdzenie oprogramowania użytkowego. Nie zauważyłem żadnych nieprawidłowości, co świadczy o tym, że system w wersji 1.3 jest poprawnie „instalowany" przez program. Oczywiście programy, które odmawiały współpracy pod kontrolą systemu w wersji 2.0, zaczynały działać — przykładem może tu posłużyć DynaCADD, nota bene reklamowany jako „2.0 compatible", który bez przeszkód „ruszył" dopiero po załadowaniu starego systemu.

Bardzo podobnie było z wszelakimi demami i grami. Z tych, które nie działały z Kickstart'em 2.0, przytłaczająca większość zaczęła „chodzić". Większość, nie wszystkie — są bowiem wypadki, w których najgenialniejszy program nic nie pomoże. Mam tu na myśli dwie możliwości: bezpośrednie skoki w obszar ROM-u i niekompatybilność wywołaną zastosowaniem w Amidze 500+ nowych wersji kostek specjalizowanych. Jeśli bowiem gra, demo czy program użytkowy napisany dla Kictstart-a 1.3 wykonuje skok w jakieś miejsce ROM-u, to Emulator 1.3 nie zapewni tu jego poprawnego działania — Kickstart 1.3 przez niego wczytany znajduje się w obszarze adresowym RAM.

Tak samo niemożliwe jest zlikwidowanie błędów wynikających z „nielegalnego" używania przez programy kości specjalizowanych Amigi — na przykład gra SWIV (a przynajmniej jej piracka wersja) zawiesza się na Amidze 500+ nawet po fizycznej wymianie ROM-u.

## PODSUMOWANIE

Firma Alderan zaczynała od niezłej jakości programów, Emulator 1.3 jest już jednak, obok innych nowych tytułów tej firmy, programem na naprawdę bardzo wysokim poziomie — całkowicie spełnia swoje zadanie, jest bardzo prosty w użyciu (czy może być coś prostszego od włożenia i wyjęcia po chwili dyskietki?) — po prostu nie ma się do czego przyczepić. Z czystym sumieniem mogę polecić Emulator 1.3 wszystkim sfrustrowanym posiadaczom Amig 500+ — komapatybilność z Amigą 500 jest bowiem doskonała.

*TOMASZ KOWALSKI*

***P.S.**: od dystrybutora programu otrzymaliśmy zapewnienie, że w ciągu najdalej kilku tygodni będzie gotowa wersja programu przeznaczona dla Amigi 600 i 2000D.*

**DYSTRYBUTOR:**
**Alderan S.C., ul. Korotyńskiego 19a/55, 02-123 Warszawa, tel. 659-18-21**

## ZALETY:

+ zapewnienie prawie idealnej kompatybilności ze „starą pięćsetką"
+ rewelacyjna prostota „obsługi"
+ stosunkowo niska cena

## WADY:

- program zajmuje zbyt dużo pamięci (256 KB)

Był okres, kiedy dostawaliśmy od Was listy z prośbami o rozpoczęcie kursu asemblera. Od momentu, kiedy zaczęły ukazywać się kolejne odcinki kursu, do redakcji nie przyszedł żaden list poruszający problem programowania w asemblerze. Spotkałem się jedynie z krytycznymi uwagami mówiącymi, że nie opisuję rozkazów asemblera, tylko od razu rozpocząłem od opisywania procedur systemu operacyjnego. Na polskim rynku dostępna jest książka pana Jacka Kostrzewskiego, która w wyczerpujący sposób opisuje asembler Motoroli 68000, dlatego też nie widziałem celu w dublowaniu wiadomości zawartych w tymże podręczniku. W dzisiejszym odcinku chciałbym zająć się troszeczkę czymś innym, chcę mianowicie zapoznać Was z koprocesorem Copper.

Na pewno wszyscy widzieli skaczące po ekranie różnokolorowe paski, czy też bajecznie kolorowe plansze cyklicznie zmieniające kolory — wszystko to zostało zrobione właśnie za pomocą Coppera. Copper jest koprocesorem, więc tak, jak prawdziwy procesor, posiada własne instrukcje. Z instrukcji możemy zbudować program, umieścić go w pamięci i uruchomić podając Copperowi jego początkowy adres. Liczba rozkazów koprocesora nie jest może imponująca, ale jego rozkazy są bardzo uniwersalne.

Copper rozpoznaje trzy rozkazy: MOVE, WAIT, SKIP; za ich pomocą możemy w pro-

STRUKTURA ROZKAZÓW COPPERA:

| MOVE | | | WAIT | | SKIP | |
|---|---|---|---|---|---|---|
| bit | S1 | S2 | S1 | S2 | S1 | S2 |
| 15 | x | D15 | V7 | BFD | V7 | BFD |
| 14 | x | D14 | V6 | VM6 | V6 | VM6 |
| 13 | x | D13 | V5 | VM5 | V5 | VM5 |
| 12 | x | D12 | V4 | VM4 | V4 | VM4 |
| 11 | x | D11 | V3 | VM3 | V3 | VM3 |
| 10 | x | D10 | V2 | VM2 | V2 | VM2 |
| 9 | x | D9 | V1 | VM1 | V1 | VM1 |
| 8 | R8 | D8 | V0 | VM0 | V0 | VM0 |
| 7 | R7 | D7 | H8 | HM8 | H8 | HM8 |
| 6 | R6 | D6 | H7 | HM7 | H7 | HM7 |
| 5 | R5 | D5 | H6 | HM6 | H6 | HM6 |
| 4 | R4 | D4 | H5 | HM5 | H5 | HM5 |
| 3 | R3 | D3 | H4 | HM4 | H4 | HM4 |
| 2 | R2 | D2 | H3 | HM3 | H3 | HM3 |
| 1 | R1 | D1 | H2 | HM2 | H2 | HM2 |
| 0 | 0 | D0 | 1 | 0 | 1 | 1 |



# ASEMBLER 68000 (IV)

sty sposób zmieniać zawartość rejestrów układów specjalizowanych Amigi w zależności od położenia wiązki elektronów na ekranie (wiązka elektronów biegnie od lewego górnego rogu do prawego dolnego; cały ekran przebiega w ciągu 1/50 sekundy). Każda z instrukcji koprocesora zapisana jest na dwóch słowach (32 bitach).

### Instrukcja MOVE

Służy do wpisywania bezpośredniej wartości do rejestrów układów specjalizowanych Amigi. Rozkazem tym możemy zmienić zawartość rejestrów umieszczonych od adresu $DFF040 do $DFF1BE. Należy jednak pamiętać, że dostęp do rejestrów *blittera* (od $DFF040 do $DFF07E) jest możliwy dopiero po ustawieniu bitu zerowego w rejestrze COPCON ($DFF02E). Rozkaz MOVE zapisany jest na dwóch słowach, z których pierwsze jest adresem rejestru (patrz tabele w C&A nr 6 i nr 7), a drugie wartością wpisywaną do tego rejestru. Jeżeli chcemy np. ustawić czerwony kolor tła ($DFF180), rozkaz MOVE w programie Coppera będzie postaci:

**dc.w $180** ; adres rejestru koloru COLOROO;
**dc.w $0f00**; słowo danych. Adres rejestru musi być zawsze wartością parzystą.

### Instrukcja WAIT

Powoduje przerwanie pracy Coppera do momentu osiągnięcia przez wiązkę elektronów podanej przez nas pozycji na ekranie. Pierwsze słowo instrukcji określa pozycję na ekranie (bit 0 tego słowa musi być zawsze ustawiony): pierwszy bajt to pozycja pionowa (od 0 do 255), drugi — pozycja pozioma. Wartość pozioma zapisana jest w dziewięciu bitach, ale możemy podać tylko górne siedem. Podawane przez nas pozycje będą w krokach co cztery punkty niskiej rozdzielczości. Przy określaniu pozycji pionowej musimy pamiętać o tym, że linii na ekranie jest 313, a my mamy tylko osiem bitów dla określenia konkretnej linii, więc rozkaz WAIT nie może odróżnić linii 0–255 od pozostałych. Jeżeli chcemy więc, aby Copper oczekiwał na linię z zakresu 256–313, musimy użyć dodatkowego rozkazu WAIT:

1. oczekującego na linię 255,
2. oczekującego na żądaną linię (linie liczone są znów od zera).

Drugie słowo rozkazu zawiera bity maski (bit 0 jest wyzerowany). Możemy określić, które bity pozycji poziomej i pionowej są aktualnie brane pod uwagę. Bit 15 słowa maski określa, czy Copper ma czekać, aż *blitter* zakończy pracę. Jeśli jest on ustawiony, to status *blittera* jest ignorowany (Copper nie czeka na zakończenie operacji *blittera*).

### Instrukcja SKIP

Sprawdza, czy aktualna pozycja jest większa lub równa od podanej w słowie komendy. Jeśli wynik będzie pozytywny to Copper pominie następny rozkaz. Przeznaczenie bitów w dwóch słowach instrukcji jest takie samo, jak w przypadku rozkazu WAIT, jedynie zerowy bit drugiego słowa musi być ustawiony.

Po stworzeniu własnego programu dla Coppera (CopperListy), możemy przystąpić do jej uruchomienia. Należy pamiętać, że koniec programu Coppera będzie wyznaczony nielegalnym rozkazem (niemożliwym do wykonania przez Coppera) np.:

**dc.w $FFFF, $FFFE**; instrukcja WAIT

Adres programu Coppera wpisuje się do rejestru COP1LC ($DFF080) i jednocześnie zeruje się zawartość rejestru COPJMP1 ($DFF088). Przed ustawieniem adresu powinniśmy wyłączyć kanał DMA dla Coppera:

**MOVE.W #$0080, $DFF096**

Włączenie kanału DMA dla Coppera spowoduje uruchomienie jego programu:

**MOVE. W #$8280, $DFF096**

Obok znajduje się przykładowy program wyświetlający na ekranie cztery kolorowe pasy.

*BARTOSZ SMAGA*

*OBJAŚNIENIA:*

***S1*** *— pierwsze słowo rozkazu*
***S2*** *— drugie słowo rozkazu*
***x*** *— bit jest nieużywany (powinien być ustawiony na 0)*
***R8-R1*** *— adres rejestru*
***D15-D0*** *— słowo danych*
***V7-V0*** *— pionowa pozycja wiązki*
***H8-H2*** *— pozioma pozycja wiązki (bity od 2 do 8)*
***BFD*** *— oczekiwanie na koniec pracy blittera — kiedy bit jest wyzerowany*
***VM6-VMO*** *— bity maski pionowej*
***HM8-HM2*** *— bity maski poziomej*

```
START:
  lea      COPPERLIST(pc),a0 ; adres cop.list
  move.w   #$01ff,$dff096    ; zamknięcie kanałów DMA
  move.l   a0,$dff080        ; ustawienie adresu cop.listy
  clr.w    $dff088           ; wyzerowanie rejestru COPJMP1
  move.w   #$8280,$dff096    ; włączenie kanału DMA dla
                             ; coppera
MOUSE:
  btst     #$6,$bfe001       ; oczekiwanie na LMB
  bne.b    MOUSE
WYJSCIE:
  move.l   $4,a6             ; EXECBASE do A6
  lea      GRlib(pc),a1      ; nazwa biblioteki w A1
  jsr      -408(a6)          ; otworzenie biblioteki
  move.l   d0,a0
  move.w   #$01ff,$dff096    ; wyłączenie kanałów DMA
  move.l   $26(a0),$dff080   ; wpisanie adresu systemowej
                             ; cop.listy
  clr.w    $dff088
  move.w   #$83ff,$dff096    ; włączenie wszystkich kanałów
                             ; DMA
  move.l   d0,a1             ; bazowy adres biblioteki w A1
  jsr      -414(a6)          ; zamknięcie biblioteki
  rts                        ; wyjście z programu

GRlib:  dc.b  'graphics.library',0 ; nazwa biblioteki
        even                        ; instrukcja ASM-ONEa
                                    ; (parzysty adres)
COPPERLIST:                  ; początek copperlisty
  dc.w    $180,$a00          ; wpisanie wartości koloru tła
  dc.w    $5009,$fffe        ; instrukcja WAIT
  dc.w    $180,$cc0
  dc.w    $a009,$fffe
  dc.w    $180,$00e
  dc.w    $f009,$fffe
  dc.w    $180,$0b0
  dc.w    $ffff,$fffe        ; instrukcja nielegalna
                             ; (koniec copperlisty)
```

**BLITTER**

Część układu Agnus służąca do przenoszenia dużych bloków pamięci. Za pomocą *blittera* można też rysować na ekranie linie i wypełniać punktami powierzchnie



# AMOS (3)

W poprzednim odcinku poznaliśmy takie elementy Amosa, jak instrukcje, zmienne, operacje arytmetyczne. Dzisiaj zajmiemy się kolejną porcją instrukcji, a także wspomnimy o etykietach.

Znamy już następujące instrukcje:

**Print** — wyprowadzanie tekstu i liczb na ekran,
**Input** — pobieranie tekstu lub liczb od użytkownika,
**If... Then...** — instrukcja umożliwiająca podjęcie przez program decyzji o dalszej pracy, w zależności od pewnego parametru.

W poprzednim odcinku stworzyliśmy mały program. Zawierał on pewną ilość instrukcji, po których wykonaniu kończył swoje działanie. Zastanówmy się jednak, co należałoby zrobić, aby program po zakończeniu działania nie zatrzymywał się, ale umożliwiał powtórzenie wszystkich operacji jeszcze raz. Z logicznego punktu widzenia należałoby jeszcze raz powtórnie wpisać wszystkie jego instrukcje. Wyglądałoby to mniej więcej tak:

```
Program XXX
Instrukcje
Koniec
Program XXX
Instrukcje
Koniec
```

Wszystko pięknie, ale co będzie, jeśli będziemy chcieli, aby program powtarzał swoje operacje kilka, kilkanaście, lub nawet kilkadziesiąt razy? Nikt nie będzie przecież tyle razy wpisywał wszystkich instrukcji. Wystarczy przecież, aby program po zakończeniu swoich działań wracał jeszcze raz do początku. Można tego dokonać za pomocą etykiet i instrukcji skoków.

Etykieta to po prostu ciąg znaków, zbudowany na takich zasadach jak zmienne całkowite (patrz odcinek poprzedni), z tym, że zakończone są dwukropkiem, np. ETYKIETA:, KONIEC:, itp. Etykietami mogą być także liczby całkowite. Zarówno jedne jak i drugie umieszczamy zawsze na początku linijki. Etykiety tworzone pierwszym sposobem mogą występować w linii samodzielnie, natomiast etykiety liczbowe muszą poprzedzać jakąś instrukcję, np,:

```
POCZATEK:
Print „Jak masz na imie czlowieku"
Input IMIE$
Print "Powiedz mi ;„IMIE$;", ile masz lat?"
Input WIEK
WARUNKI:
If WIEK<18
   Print "Och „;IMIE$;", jestes jeszcze nie pelnoletni!!!"
Else
   Print "Witamy w naszym programie panie „;IMIE$
End If
KONIEC:
End
```

Instrukcja "End", choć jest tutaj zbędna, powoduje zakończenie wykonywania programu. A teraz to samo drugim sposobem:

```
10 Print "Jak masz na imie czlowieku"
20 Input IMIE$
30 Print "Powiedz mi „;IMIE$;", ile masz lat?"
40 Input WIEK
50 If WIEK<18
   Print "Och „;IMIE$;", jestes jeszcze niepelnoletni!!!"
60 Else
   Print "Witamy w naszym programie, panie „;IMIE$
70 End If
80 End
```

Przypomina to bardzo programy pisane w Basic „małych" komputerów. I tak jest w istocie. Interpretery Basic tych maszynek wymagają liczbowej etykiety na początku każdej linii. My jednak mamy tu sporą działalność. Numery linii mogą być umieszczone w każdym miejscu programu, może też ich wcale nie być, a numery etykiet nie muszą być ułożone rosnąco. Jedynym ograniczeniem jest to, że nie mogą się one powtarzać. Dlaczego? O tym za chwilę, a teraz wrócimy do postawionego na początku problemu.

Widzimy doskonale, że obecność etykiet nie wpływa na wykonywanie programu. Po co więc one są? Zaznaczają one pewne wybrane przez nas miejsca, gdzie możemy skoczyć z innego miejsca za pomocą instrukcji, którą teraz poznamy.

## Goto ...

Jest to instrukcja powodująca skok programu do etykiety zawartej w parametrze, np. „Goto ETYKIETA", „Goto 10", itp,. z tym, że po nazwie etykiety tworzonej „pierwszym" sposobem nie umieszczamy już dwukropka.

A teraz zmodyfikujemy nasz program tak, aby po skończeniu wykonywania wszystkich instrukcji wracał do początku.

```
POCZATEK:
Print "Jak masz na imie czlowieku"
Input IMIE$
Print "Powiedz mi „;IMIE$;", ile masz lat?"
Input WIEK
WARUNKI:
If WIEK<18
   Print "Och „;IMIE$;", jestes jeszcze niepelnoletni!!!"
Else
   Print "Witamy w naszym programie panie „;IMIE$
End If
KONIEC:
Goto POCZATEK
```

a w drugim przypadku:

```
10 Print "Jak masz na imie czlowieku"
20 Input IMIE$
30 Print "Powiedz mi „;IMIE$;", Ile masz lat?"
40 Input WIEK
50 If WIEK<18
   Print "Och „;IMIE$;", jestes jeszcze niepelnoletni!!!"
60 Else
   Print "Witamy w naszym programie, panie „;IMIE$
70 End If
80 Goto 10
```

Ze zrozumiałych względów program nie będzie posiadał instrukcji "End".

Tak więc już wiemy, jak wykonuje się skoki. Kombinacja instrukcji warunkowych i skoków pozwala programowi na rozbudowane obsługiwanie różnych skomplikowanych procesów, podejmowanie decyzji o dalszym działaniu, itp.

Wróćmy jeszcze raz do naszego programu. Każdy porządny program powinien mieć możliwość zakończenia pracy. Nie będziemy przecież resetować całego systemu, aby wyjść z tak prostego programu (jest to ogólna uwaga mająca

**DEBUGGER**

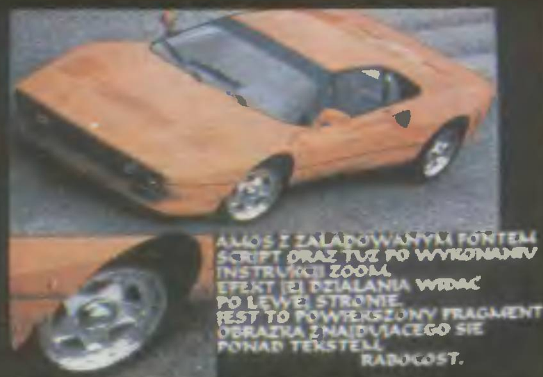
Specjalny program umożliwiający wykonywanie napisanego programu krok po kroku, w celu łatwego zlokalizowania błędów. Instrukcja aktualnie wykonywana jest wyświetlana na ekranie. Po naciśnięciu jakiegoś klawisza, zwykle spacji, wykonywana jest następna instrukcja. W Amosie, jeżeli w parametrze instrukcji „Follow" umieścimy nazwę jakiejś zmiennej (może być ich więcej, oddzielamy je wtedy przecinkiem) będziemy mogli zaobserwować, jak w trakcie działania programu zmienia się jej wartość.



*Fraktal zrobiony za pomocą Amosa*








*Takie sztuczki też dsją się zrobic „na AMOSIE"...*

na celu wprowadzanie od początku dobrych nawyków — program w Amosie możemy bowiem przerwać w każdej chwili kombinacją klawiszy Ctrl+C). Aby otrzymać ostateczną wersję, dopisz nowe zakończenie (wyrzuć z poprzedniego programu instrukcję „Goto POCZĄTEK" po etykiecie KONIEC):

```
KONIEC:
Print "To by bylo na tyle. Jeszcze raz? (t/n)"
Input PYT$
If PYT$="t"
  Goto POCZĄTEK
Else
  If PYT$="n"
    End
  Else
    Print "Odpowiedz ,t' lub ,n' "
    Goto KONIEC
  End If
Endt If
```

A teraz zastanówmy się, dlaczego etykiety nie mogą być zdublowane. Po tym, co napisałem o instrukcjach skoków nasuwa się prosty wniosek: komputer musi mieć dokładnie określone, gdzie nastąpi skok — nie może przecież skoczyć do dwóch miejsc naraz.

Poznaliśmy więc podstawową instrukcję umożliwiającą skok. Chciałbym jednak oznajmić, iż nie jest to jedna instrukcja pozwalająca zmienić miejsce wykonywania programu. Wyobraźmy sobie, że jakiś stały fragment napisanego przez nas programu powtarza się kilka razy, np.:

```
POCZATEK:
Print "Jak ma na imie twoja matka?"
Input MATKA$
Print "Ile ma lat?"
Input MATKA
Print "Jak ma na imie twoj ojclec?"
Input OJCIEC$
Print "Ile ma lat?"
Input OJCIEC
Print "Jak ma na imie moj szanowny uzytkownik?"
Input TY$
Print "Ile masz lat?"
Input TY
ETYKIETA:
Prlnt "Teraz poznalem cala wasza rodzine."
Print "Ty nazywasz się „;TY$;" i jestes mlodszy"
Print "od matki o";MATKA-TY;" lat(a), a od"
Print "ojca o";OJCIEC-TY;" lat(a),"
Print "Twoja matka — „;MATKA$;" jest"
If MATKA<>OJCIEC
  If MATKA>OJCIEC
    Prlnt "starsza";
  Else
    Print "mlodsza";
  End If
  Print "od meza o imieniu ";OJCIEC$
  Prlnt "o";Abs (MATKA-OJCIEC);" lat(a)."
Else
  Print "w tym samym wieku co jej maz — ";OJCIEC$
End If
KONIEC:
End
```

Instrukcja **Abs (...)**, którą dokładniej zajmiemy się przy innej okazji, oblicza wartości bezwzględne z parametru liczbowego umieszczonego w nawiasie.

Proszę zwrócić uwagę, że cały pierwszy człon programu (od etykiety POCZĄTEK do ETYKIETA) to stale powtarzający się tekst i pytanie o czyjeś imię i wiek. Można byłoby więc zastąpić potrójne pytanie jednym tzw. podprogramem i trzy razy się do niego odwołać. Problem polega jedynie na tym, że różnią się między sobą teksty wyprowadzane instrukcją „Print" (chodzi o pytanie o czyjeś imię, bo pytania o wiek są takie same). Rozwiązemy to za pomocą dodatkowej zmiennej. Problemem jest także powrót z naszego podprogramu do programu głównego (wywołanie jest proste — skok do etykiety oznaczającej początek podprogramu, ale powrót będzie musiał odbywać się w trzy różne miejsca). Do tego właśnie celu służy para instrukcji **Gosub — Return**. Najpierw jednak popatrzmy na program.

```
POCZATEK:
IMIE$="twoja matka" : Gosub PODPROGRAM
MATKA$=KTOS$ : MATKA=KTOS
IMIE$="twoj ojciec" : Gosub PODPROGRAM
OJCIEC$=KTOS$ : OJCIEC=KTOS
IMIE$="moj szanowny uzytkownik" : Gosub PODPROGRAM
TY$=KTOS$ : TY=KTOS
ETYKIETA:
Print "Teraz poznalem cala wasza rodzine."
Print "Ty nazywasz sie „TY$;" i jestes mlodszy"
Print "od matki o";MATKA-TY;" lat(a), a od"
Print "ojca o";OJCIEC-TY;" lat(a)."
Print "Twoja matka — ";MATKA$;" jest"
If MATKA<>OJCIEC
  If MATKA>OJCIEC
    Print "starsza";
  Else
    Print "mlodsza";
  End If
  Print "od meza o imieniu ";OJCIEC$
  Print "o";Abs(MATKA-OJCIEC);" lat(a)."
Else
  Print "w tym samym wieku co jej maz — ";OJCIEC$
End If
KONIEC:
End
PODPROGRAM:
Print "Jak masz na imie ";IMIE$;"?"
Input KTOS$
Print "Ile ma lat?"
Input KTOS
Return
```

Jak widzimy podprogram znajduje się na końcu właściwego programu i jest wywoływany przez instrukcję „Gosub PODPROGRAM". Taki skok różni się od wcześniej poznanego „Goto ..." tym, że zapamiętuje w programie miejsce, z którego został wykonany. Dalej instrukcje wykonują się normalnie, aż do napotkania instrukcji „Return". Powoduje to powrót do pierwszej instrukcji, która znajduje się po instrukcji „Gosub". Pozwala to właśnie na budowanie podprogramów, czyli grup instrukcji, które możemy wywołać w dowolnym momencie programu i nie martwić się o powrót — resztę wykona komputer. Takie programowanie przypomina budowanie z klocków i pozwala zachować na listingu porządek. Niesie także ze sobą najważniejszą pozytywną cechę: jest bliższe programowaniu strukturalnemu. Jeszcze lepsze byłyby tu procedury, ale o tym później. Na razie proponuję pisanie prostych programów, które wcale nie muszą przedstawiać sobą jakiejś większej wartości użytkowej (najlepszym tego przykładem są powyższe listingi). Chodzi nam po prostu o obeznanie się z językiem, nauczenie się komputerowego, nie zawsze logicznego dla nas „myślenia".

Na koniec chciałbym przedstawić jeszcze jedną instrukcję Amosa:

## Follow

Użycie jej powoduje włączenie czegoś w rodzaju *debuggera* (wyjaśnienie tego terminu w ramce), co może być nieocenioną pomocą dla każdego początkującego programisty. Np. w razie niezrozumienia działania jakiegokolwiek z programów umieszczonych na listingach można je dokładnie, krok po kroku prześledzić. Użycie instrukcji **Follow Off** powoduje wyjście z *debuggera* i powrót do normalnej pracy Amosa.

I to by było na tyle. Gdyby pojawiły się jakieś większe problemy, proszę o listy, odpowiem chętnie na wszystkie pytania.

(cdn.)

*RAFAŁ BORZYŃSKI (RABOCOST)*





# BEZPOŚREDNIE ODCZYTYWANIE DANYCH

Jeśli chcecie wiedzieć, jak odczytać dane z dysku bezpośrednio ze ścieżek nie korzystając z DOS-u, to przeczytajcie uważnie ten artykuł.

Najpierw słówko o strukturze dysku w Amidze: 80 ścieżek (tracks) po 22 sektory każda (po 11 sektorów na każdej stronie dyskietki). Długość jednego sektora to 512 ($200) bajtów.

Za obsługę dysku na poziomie ścieżek odpowiada trackdisk.device. Jak każde urządzenie, wymaga ono otwarcia. Służy do tego procedura OpenDevice (offset -414, biblioteka Exec). Jako dane wejściowe podajemy: w D0 — unitNumber (zależy od urządzenia), w D1 — flags (też zależne od urządzenia), w A0 — adres nazwy (czyli trackdisk.device), i w A1 — adres struktury IOStdReq. W naszym przypadku D0 i D1 będą wyzerowane. Funkcja ta w D0 zwraca nam kod błędu. Jeśli jest on równy 0, to znaczy, że wszystko poszło dobrze. Wywołanie z C wyglądałoby tak:

error=OpenDevice(devName,unitNumber,IOStdReq,flags).

Po zakończeniu zabawy z urządzeniem, powinniśmy koniecznie je zamknąć procedurą CloseDevice (-450,Exec), której jedynym argumentem jest adres struktury IOStdReq w rejestrze A1. W C: CloseDevice(IOStdReq).

Wypadałoby teraz opisać wspomniane struktury.

**IOStdReq**

| | | |
|---|---|---|
| **$0** | **s** | **Message io_Message** |
| **$14** | **1** | **Device *io_Device** |
| **$18** | **1** | **Unit *io_Unit** |
| **$1c** | **w** | **io_Command** |
| **$1e** | **b** | **io_Flags** |
| **$1f** | **b** | **io_Error** |
| **$20** | **1** | **io_Actual** |
| **$24** | **1** | **io_Length** |
| **$28** | **1** | ***io_Data** |
| **$2c** | **1** | **io_Offset** |

Jak widać, zawiera ona w sobie strukturę Message.

**Message**

| | | |
|---|---|---|
| **$0** | **s** | **Node** |
| **$e** | **l** | **MsgPort *mn_ReplyPort** |
| **$12** | **w** | **mn_Length** |

Brakuje jeszcze opisu struktury MsgPort:

**MsgPort**

| | | |
|---|---|---|
| **$0** | **s** | **Node mp_Node** |
| **$e** | **b** | **mp_Flags** |
| **$f** | **b** | **mp_SigBit** |
| **$10** | **l** | **Task *mp_SigTask** |
| **$14** | **s** | **List mp_MsgList** |

**i Node**

| | | |
|---|---|---|
| **$0** | **l** | **Node *ln_Succ** |
| **$4** | **l** | **Node *ln_Pred** |
| **$8** | **b** | **ln_Type** |
| **$9** | **b** | **ln_Pri** |
| **$a** | **l** | ***ln_Name** |

Zanim wywołamy OpenDevice, musimy ustawić pewne dane w strukturach. Właściwie wystarczy w MsgPort, w polu mp_SigTask podać adres struktury naszego *tasku* (zwraca go w D0 procedura FindTask (-294, Exec), jeżeli w A1 podamy 0), a w IOStdReq, w mn_ReplyPort adres struktury MsgPort. Jest to zestaw minimum, a można jeszcze wypełnić pola ln_Type dla obu struktur (jako 4 dla MsgPort, 5 dla Message), ustawić priorytet w ln_Pri dla IOStdReq, nazwę MsgPort'u w ln_Name. Komunikacja z urządzeniem odbywa się za pomocą funkcji DoIO (-456, Exec), której argumentem jest adres tejże właśnie struktury IOStdReq w A1. Jest to uniwersalny sposób wykorzystywania urządzeń systemowych. Różnice pomiędzy urządzeniami polegają na interpretowaniu danych Command, Flags, Actual, Lenght, Data i Offset.

Standardowych komend jest 9 i są nimi:

**CMD_INVALID 0** — zła komenda
**CMD_RESET 1** — reset urządzenia
**CMD_READ 2** — standardowe czytanie
**CMD_WRITE 3**— standardowy zapis
**CMD_UPDATE 4** — zapisz wszystkie bufory
**CMD_CLEAR 5** — wyczyść wszystkie bufory
**CMD_STOP 6** — zatrzymanie aktualnej operacji
**CMD_START 7** — wznowienie operacji
**CMD_FLUSH 8** — definitywne przerwanie operacji
**CMD_NONSTD 9** — pierwsza komenda niestandardowa

Są to komendy prawdziwe dla każdego urządzenia. Ponieważ jednak poszczególne urządzenia mogą potrzebować specjalnych funkcji, wprowadzono komendy niestandardowe, które mają numery zaczynające się od liczby 9. Dla trackdisc.device są to:

**TD_MOTOR 9** — kontrola silnika stacji
**TD_SEEK 10** — seek (do testowania)
**TD_FORMAT 11** — formatowanie dysku
**TD_REMOVE 12**
**TD_CHANGENUM 13**
**TD_CHANGESTATE 14** — czy jest dysk w stacji?
**TD_PROTSTATUS 15** — czy dysk jest zabezpieczony przed zapisem
**TD_RAWREAD 16**
**TD_RAWWRITE 17**
**TD_GETDRIVETYPE 18** — pobierz typ dysku
**TD_GETNUMTRACKS 19** — pobierz ilość ścieżek na dysku
**TD_ADDCHANGEINT 20**
**TD_REMCHANGEINT 21**
**TD_LASTCOMM 22**

Dość teoretyzowania, pora na konkrety. Żeby odczytać dane z dysku, jako io_Command musisz podać 2 (read), io_Length wielkość wczytywanych danych (powinno być wielokrotnością 512, gdyż na raz wczytany może być NAJMNIEJ jeden sektor), w io_Data adres bufora, do którego wczytujemy dane (oczywista, musi być co najmniej tak duży, jak długość wczytywanych danych) i w io_Offset offset, czyli odległość od początku dysku. Offset danego sektora oblicza się bardzo prosto:

offset=512*numer sektora. Teraz tylko DoIO i dane powinny zostać wczytane. Nagrywanie na dysk wygląda niemal identycznie, tylko jako io_Command podać trzeba 2 (write). Należy pamiętać, że offset i długość zawsze powinny być wielokrotnością 512, a bufor koniecznie znajdować się w CHIP MEM.

Jeśli zrobiłeś kilka eksperymentów z odczytywaniem, zauważyłeś zapewne, że nawet po zakończeniu operacji pali się dioda stacji. Czytanie i zapis nie są kończone wyłączeniem motoru stacji i trzeba to zrobić własnoręcznie. Służy do tego komenda TD_MOTOR (9). W io_Length podajemy 0 (gdy chcemy wyłączyć motor) lub 1 (gdy chcemy go włączyć). Po wykonaniu operacji, z io_Actual możemy odczytać poprzedni stan motora. Ponieważ nic tak nie wyjaśnia sprawy jak krótki przykład, zamieszczam listing programu w asemblerze, który odczytuje *bootblock* (czyli 2 pierwsze sektory na dysku).

Przydatnymi komendami są jeszcze: TD_CHANGESTATE (14), zwracająca stan dysku w io_Actual (gdy jest to wartość niezerowa — brak dysku w stacji), TD_GETNUMTRACS (19), w io_Actual zwracająca ilość ścieżek dostępnych danemu typowi dysku oraz TD_PROTSTATUS (15), sprawdzająca zabezpieczenie przed zapisem (gdy io_Actual<>0 – > dysk jest zabezpieczony).

Na koniec jeszcze jedna informacja dla piszących programy dla *bootblocków*: nie musisz się kłopotać otwieraniem trackdisk.device, gdyż system wykonując program na *bootblocku*, przekazuje mu w rejestrze A1 adres kompletnej struktury IOStdReq!

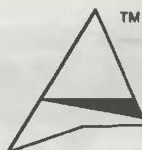
A teraz obiecany przykład (sprawdzony pod Seką, AsmOne i DevPackiem).

***KOVI***

```
OpenDevice        = -444
CloseDevice       = -450
DoIo              = -456
FindTask          = -294
ln_Type = $8
ln_Pri  = $9
mn_ReplyPort = $e
mp_SigTask   = $10
io_Command   = $1c
io_Length    = $24
io_Data      = $28
io_Offset    = $2c

Start:  bsr     OpenTrackDisk
        bne     Exit
        bsr     Read
        bsr     MotorOff
        bsr     CloseTrackDisk
Exit:   rts

OpenTrackDisk:
        move.l  4,a6
        sub.l   a1,a1
        jsr     FindTask(a6)     ;w D0 zwraca adres
                                  naszego zadania (task)
        lea     MsgPort(pc),a1             ;adres struktury
        move.l  d0,mp_SigTask(a1)          ;task
        move.b  #4,ln_Type(a1)             ;typ:MsgPort
        lea     IORequest(pc),a1
        move.b  #5,ln_Type(a1)             ;typ:message
        move.l  #MsgPort,mn_ReplyPort(a1)       ;port
        move.l  4,a6
        lea     DevName(pc),a0
        moveq.l #0,d0
        moveq.l #0,d1
        jsr     OpenDevice(a6)
        tst.l   d0
        bne     ErrorTrack       ;błąd
        moveq   #0,d0
        rts
ErrorTrack:
        moveq   #-1,d0
        rts
CloseTrackDisk:
        move.l  4,a6
        lea     IORequest(pc),a1
        jsr     CloseDevice(a6)
        rts
Read:   lea     IORequest(pc),a1
        move.w  #2,io_Command(a1)
        move.l  #$400,io_Length(a1)
        move.l  #BootBlock,io_Data(a1)
        move.l  #0,io_Offset(a1)
        move.l  4,a6
        jsr     DoIO(a6)
        rts
MotorOff:
        lea     IORequest(pc),a1
        move.w  #9,io_Command(a1)
        clr.l   io_Length(a1)
        move.l  4,a6
        jsr     DoIO(a6)
        rts
        even
IORequest:      dcb.b 48,0   ;strukt. IoStdReq
MsgPort:        dcb.b 34,0   ;strukt. MsgPort
BootBlock:      dcb.b $400,0
DevName:        dc.b 'trackdisk.device',0
```



**S.C. Alderan**

Alderan S.C.
Korotyńskiego 19a/55, 02-123 Warszawa
telefon: 659-18-21

fax: 628-04-10

## S.C. Alderan prezentuje Państwu:

# *English Teacher*

- nasz najnowszy program, przeznaczony dla osób, które pragną dobrze i wszechstronnie opanować język angielski. Program zbudowany jest z kilku modułów:

**nauki słownictwa, tłumaczenia zdań, ich dokańczania oraz układania pytań**

Każdy z modułów umożliwia wybór jednego z dziesięciu stopni trudności, możliwy jest też wybór tematyki, jak na przykład komputer, zakupy, rodzina, itd. Oprócz tego wszystkiego **English Teacher** oferuje możliwość rozwiązywania angielskich krzyżówek, których jest kilkanaście! Dodatkowo, po każdej poprawnie rozwiązanej krzyżówce program wyświetla hasło, gdy zaś uczący się zbierze je wszystkie - może wysłać je na nasz adres i wziąć udział w losowaniu nagród.

W trosce o uatrakcyjnienie nauki zdecydowaliśmy się także dołączyć do **English Teacher'a** trzy w pełni wartościowe gry: Pacman, Tetris i Puzzle, będące nagrodami za poprawne rozwiązanie testów.

cena promocyjna: 108.000 zł

wymagania sprzętowe: minimum

Prowadzimy detaliczną sprzedaż wysyłkową (odbiorca płaci przy odbiorze, do ceny doliczamy koszty ponoszone na rzecz Poczty). Każdy, kto zamówi tą drogą więcej niż trzy tytuły, otrzyma gratis *Anty-Virus*, a powyżej czterech - program *Notes*.

Poszukujemy poważnych firm chcących zostać naszymi dealerami. Oferujemy korzystne warunki.

Zainteresowani jesteśmy również w nawiązaniu kontaktów ze zdolnymi programistami, a także grafikami, szczególnie specjalizującymi się w animacji trójwymiarowej.

## Poza English Teacher'em oferujemy także inne, najwyższej jakości programy użytkowe i edukacyjne:

* *A-Word 2.0* - nowa, napisana praktycznie od nowa wersja najpopularniejszego słownika angielsko-polskiego dla Amigi. Wszystkie funkcje wersji 1.0 (praca w multitaskingu, niewiarygodnie krótki czas wyszukiwania słów, trzy tryby pracy, specjalny słownik terminów związanych z Amigą), a ponadto szereg nowych: pełna synteza mowy, rozbudowana współpraca z drukarką, nowy, wspaniały interface użytkownika, możliwość wpisywania słów do tłumaczenia na kilka sposobów (w tym "ze słuchu", czyli w takiej formie, w jakiej słowo usłyszeliśmy!), blisko dwukrotnie większa baza danych niż w wersji 1.0, a do tego wszystkiego jeszcze możliwość pracy w specjalnym nowym trybie jako słownik polsko-angielski!
[cena promocyjna: 120.000 zł]

* *Statystyka* - najwyższej jakości program użytkowy o pełnej skali zastosowań: od budżetu domowego po profesjonalną statystykę. Szesnaście typów wykresów, pełny WYSIWYG, opcja dla studentów (zakres pierwszych dwóch lat studiów ze statystyki!), możliwość dokonywania porównań dwóch szeregów, praca w standardzie IFF, edytor graficzny pomagający w uzupełnianiu wykresów o dodatkowe informacje, interpretacja uzyskanych wyników, skalowanie gotowych wykresów, możliwość tworzenia bazy szeregów zdefiniowanych w programie. *Statystyka* została zaakceptowana przez Politechnikę Gdańską jako narzędzie nauki dla studentów!
[cena promocyjna: 160.000 zł]

* *Avalon* - doskonała, wciągająca gra logiczna. Dobra dla każdego - dla szukającego ciekawej rozrywki dorosłego i dla rozwijającego się dziecka. Gwarantuje to bardzo duże zróżnicowanie stopnia trudności w poszczególnych etapach gry.
[cena promocyjna: 80.000 zł]

***Nowości***

* *Poliglota (Lernt mit uns)* - doskonały program do nauki języka niemieckiego, oparty na najlepszych, powszechnie uznanych materiałach. Poliglota umożliwia naukę słów, wyrażeń i zwrotów w obu "kierunkach", czyli na pytanie o niemieckie słowo, wyrażenie lub zwrot możemy odpowiadać po polsku albo na polskie - po niemiecku. Materiał podzielony jest na zestawy, każdy po kilkadziesiąt pól, zaś samych zestawów jest 216! Użytkownik ma też możliwość samodzielnego definiowania nowych zestawów. *Poliglota* wyposażony jest też w syntetyzer mowy, co, zważywszy, że poprawnie wymawia on słowa niemieckie, jest rewelacją!
[cena promocyjna: 90.000 zł]

* *Zestaw biznesowy* - fakturowanie, kosztorys, księgowość, magazyn, kadry, płace - wszystko w jednym zestawie. Obliczanie wszelkich podatków, wydruk faktur - tak jak na IBM PC, tylko kilkukrotnie taniej. Wyczerpujący opis - na życzenie. Zapewniamy pełny serwis, łącznie ze sprzedażą kompletnych systemów.
[cena promocyjna do końca roku: 1.500.000 zł]

* *Panda Pakiet Matematyczny* - składa się z kilku programów. *Panda* - rozwiązywanie równań, układów równań dwiema technikami, rozkład wielomianów, działania na wielomianach, wszelkie informacje o funkcji kwadratowej (miejsca zerowe, wyróżnik, wierzchołek paraboli, monotoniczność itd.). Program *Funkcje* - różnokolorowe, nakładane jeden na drugi wykresy, analiza wykresów funkcji, kreślenie i wyliczanie pochodnej, aproksymacja pierwiastków, całkowanie (!), wbudowane stałe matematyczne. *Matematyka* - pola, obwody figur, wykresy statystyczne, układy równań, macierze, silnia, rachunek prawdopodobieństwa, trójkąt Pascala, działania na wielomianach, krzywe Lissajous, całki, różniczki.
[cena promocyjna 85.000 zł]

* *Emulator 1.3* - rewelacja. Za ułamek ceny przeróbki hardware'owej mogą Państwo sprawić, że Wasza Amiga 500 Plus będzie w pełni kompatybilna z poprzednimi modelami. Dzięki emulacji systemu operacyjnego w wersji 1.3, znikną wszystkie Wasze kłopoty z uruchamianiem programów. Zgodność programowa Amigi Plus z uruchomionym Emulatorem 1.3 jest stuprocentowa dzięki temu, że program całkowicie wyłącza system operacyjny w wersji 2.0 i zastępuje go pełnosprawnym Kickstartem 1.3!
[cena 100.000 zł]

* *Twój Pierwszy Angielski* - wspaniała nauka języka angielskiego dla dzieci. Na program składa się 11 scen, w których dziecko ma za zadanie rozpoznać m. in. owoce, kolory i liczby. Nauka odbywa się z wykorzystaniem animacji komputerowej i syntezy mowy, zaś na końcu uczeń przystępuje do egzaminu.
[cena 160.000 zł]

* *WordTeacher 2.0* - wersja 2.0 dobrze znanego i wypróbowanego programu do nauki języka angielskiego (pisowni i wymowy). Posiada wbudowane dwa pełnosprawne słowniki: polsko-angielski i angielsko-polski (35 tysięcy słów). WT 2.0 wykorzystuje syntetyzer mowy, co umożliwia maksymalnie wierne odtwarzanie wymowy angielskiej. Dzięki metodom nauki zastosowanym w tym programie możliwe jest opanowanie z jego pomocą nawet 170 słów w ciągu godziny!
[cena 95.000 zł]

* *Ortografia (Gra Słów)* - zestaw czterech gier rozwijających wyobraźnię, spostrzegawczość, a przede wszystkim wiedzę z dziedziny ortografii (wbudowany słownik zawiera 10.000 słów prosto ze Słownika Ortograficznego). Program został skonstruowany z uwzględnieniem wszelkich reguł nauczania, nie jest, na przykład, możliwe uzyskanie na ekranie wyrazu błędnie napisanego - w pamięci utrwala się tylko poprawna pisownia.
[cena 70.000 zł]

* *Geometria Konstrukcyjna* - wszystko o geometrii, nauka wszelkich twierdzeń, sposobów rozwiązywania zadań, wszystko w oparciu o przykłady, z wykorzystaniem animacji komputerowej.
[cena 80.000 zł]

* *Pierwsze Kroki* - program zawierający kilkadziesiąt połączonych z tekstami rysunków, wyjaśniających obsługę Amigi, oprogramowania systemowego oraz sposób podłączania urządzeń zewnętrznych. Zawiera też wyjaśnienia kilkudziesięciu terminów związanych z Amigą. Idealny dla początkujących Amigowców, jak również dla firm sprzedających Amigi.
[cena 55.000 zł]

* *Chemia 2.0* - nowa, znacznie rozbudowana w stosunku do poprzedniej, wersja programu zawierającego wiadomości z zakresu chemii nieorganicznej. Znajdą tu Państwo w formie graficznej wszelkie informacje z tablicy Mendelejewa, jak również wiadomości z dziedziny mechaniki kwantowej. Program umożliwia automatyczne wyszukiwanie wszelkich zależności i podobieństw grup pierwiastków.
[cena 60.000 zł]

* *Anty-Virus* - zestaw najlepszych program antywirusowych Public Domain, jakie kiedykolwiek napisano. W zestawie znajdują się między innymi tak znane anytywirusy jak *VirusZ, BootX i VirusChecker.* Dzięki użyciu modemu jesteśmy w stanie oferować Państwu najnowsze wersje tych programów już w kilka dni po wypuszczeniu ich przez autorów.
[cena 55.000 zł]

* *Notes* - bardzo wygodna, elastyczna w swej konstrukcji i prosta w obsłudze podręczna baza danych.
[cena 200.000 zł]

* *Biorytmy 2.0* - program ten powie Państwu wszystko o Waszej kondycji psychicznej, fizycznej oraz intelektualnej.
[cena 40.000 zł]

* *Piórko* - prosty w obsłudze, ale o dużych możliwościach, dedykowany głównie dzieciom program graficzny.
[cena 60.000 zł]

***

Wszystkie programy dostępne są na komputery Amiga. Dodatkowo *Twój Pierwszy Angielski* istnieje także w wersji dla Commodore 64 (dysk i kaseta).

***

**Uwaga:** podane ceny obowiązują od dnia 01-11-92

---

***Zapraszamy***

# ZOSTAŃ WŁAMYWACZEM! (3)

## ENERGIA

Istnieje spora ilość gier, w których „kondycja" naszego bohatera zobrazowana jest kolorowym paskiem. Długość tegoż paska stopniowo zmniejsza się w miarę popełniania kolejnych błędów. Innymi słowy, bohater traci energię. Oczywiście wskaźnik energii może dotyczyć np. ilości amunicji, paliwa, powietrza itd.

Zabezpieczenie się przed utratą energii, a co za tym idzie — utratą „życia", jest dość trudne, ponieważ rozpoczynając monitorowanie pamięci nie wiemy, jaką wartość ma bajt przechowujący informację o aktualnym poziomie energii. Grafika np. kolorowego paska obrazującego energię z reguły tworzona jest przy wykorzystywaniu trybów adresowania operujących rozkazami indeksowanymi. Warto również mieć na uwadze, że podobnie jak w przypadku „żyć", sugestywnie zmniejszający się biały pasek może w rzeczywistości być „traktowany" przez program jako rosnący i czarny. Generalnie rzecz biorąc przed przystąpieniem do poszukiwań trzeba zgromadzić jak najwięcej informacji o „nieprzyjacielu".

Szczególnie interesować nas będzie czy:

— pasek energii jest ciągły czy podzielony, a jeżeli tak, to na ile części?

— ubywające porcje energii są zawsze takie same, czy zależą od obiektu, z którym zderzył się nasz bohater?

— energia tylko maleje, czy też w miarę upływu czasu następuje jej wzrost do poziomu początkowego?

— całkowita utrata energii powoduje koniec gry, czy tylko utratę „życia"?

— nie występują jakieś inne niż powyżej wymienione charakterystyczne cechy, które mogą być pomocne w odszukaniu podprogramu „energii"?

Przy pewnym doświadczeniu precyzyjne i właściwie przełożone na język asemblera odpowiedzi na powyższe pytania pozwolą na odszukanie interesujących nas informacji. Jak to zrobić zademonstruję na przykładzie gry VENOM STRIKES BACK.

Łatwo znajdujemy adres startowy $0CDE. Mamy 3 „życia". Kilka prób i zmiana DEC $D5 na LDA $D5 daje „nieśmiertelność". Gra jest dość trudna, a dodatkową trudność stanowi malejąca energia, która po wyczerpaniu się, co następuje jak zwykle tuż przed końcem kolejnego etapu, powoduje utratę „życia" i powrót do punktu wyjścia. Przechodzimy więc znów do monitora i spostrzegamy, że tuż za rozkazem DEC $D5 mamy rozkaz BPL $0D20 co oznacza, że jeżeli zawartość komórki $D5 jest nieujemna, to program powróci do adresu $0D20. Logiczne, że pomiędzy adresem startu a adresem $0D20 musi znajdować się podprogram, który „zajmuje" się energią. Nie ma innego wyjścia niż przebadanie, jaki efekt przyniesie wyłączanie po kolei wszystkich podprogramów znajdujących się w tym przedziale adresowym.

Praktycznie wykonuje się to przez wpisanie rozkazu RTS w adres, jaki wskazuje podprogram. Oczywiście, aby nie wgrywać w kółko gry, należy najpierw dokonać *dis-asemblacji* i sprawdzić, jaki jest pierwszy rozkaz w podprogramie, na miejsce którego wpisujemy RTS. Dzięki temu powrót do wersji bazowej programu nie przedstawia żadnego problemu. Mówiąc szczerze, jest to zajęcie dość monotonne, ale innego sposobu nie ma.

W końcu przychodzi kolej na sprawdzenie umieszczonego pod adresem $2F10 podprogramu JSR $2F10. Pod adresem $2F10 znajduje się kolejny podprogram (JSR $30E2), a za nim jeszcze kilka następnych podprogramów, które również musimy cierpliwie zbadać. Jako kolejny badamy podprogram JSR $3242. Pod adres $3242, gdzie figuruje rozkaz LDA $0CDB wpisujemy RTS, wracamy do Basic-a, restartujemy grę i okazuje się, że uzyskaliśmy więcej niż trzeba. Energii co prawda nie ubywa, gdyż pociski przechodzą przez naszego bohatera jak przez powietrze, ale nie może on zbierać porzuconych tu i tam elementów swojego wyposażenia. Przywracamy poprzedni rozkaz w komórce $3242 i sprawdzamy, co się znajduje w JSR $3242. Dostrzegamy w nim kolejny podprogram JSR $3268, który zaczyna się rozkazem LDA $A7,X. Wyłączamy z kolei JSR $3268 i obserwujemy efekt. Niestety dalej to samo. Konsekwentnie bierzemy się za przeglądanie podprogramu JSR $3268, gdzie począwszy od adresu $32EA znajdujemy:

```
32EA LDA $0CD4     ;zawartość komórki $0CD4 do akumula-
                    tora
32ED SEC           ;znacznik C=1
32EE SBC $3379,Y   ;od akumulatora odjąć zawartość ko-
                    mórki $3379,Y
32F1 STA $0CD4     ;rezultat przechowany w komorce
                    $0CD4.
```




Coś nam to przypomina, nieprawdaż? Rzeczywiście, udało się nam odszukać „serce" podprogramu zawiadującego poziomem energii. Cztery kolejne rozkazy NOP umieszczone w pamięci począwszy od adresu $32ED dają upragniony efekt. Dodatkowo dowiedzieliśmy się, że w komórce $0CD4 przechowywany jest aktualny poziom energii. Informacja ta, po przeszukaniu całej pamięci na obecność sekwencji bajtów D4 0C pozwoli już bez trudności ustalić, że w adresy $32E2 i $42D8 należy wpisać rozkaz SBC #$00, aby dodatkowo uzyskać odporność na pociski armatnie i miny.

W grze VENOM STRIKES BACK jest jeszcze do odszukania parę innych „usprawnień". Mam nadzieję, że Czytelnik wczuł się trochę w niełatwy problem energii, a powyższy przykład będzie pomocny w samodzielnej pracy.

## CZAS

Niemało gier wyposażono w wewnętrzny mechanizm obliczający, ile jeszcze czasu pozostało do ukończenia gry. Działa to dość deprymująco i na pewno nie ułatwia gry. Patrząc na problem szerzej będziemy mieli do czynienia z umownym „czasem" wszędzie tam, gdzie program kontroluje ubytek (najczęściej) lub przychód (rzadko) jednego z atrybutów, w które wyposażony jest nasz bohater (niezależnie od poziomu energii lub ilości żyć). Sztywny podział jest niemożliwy, gdyż np. amunicję można zaliczyć do kategorii energii lub czasu, zależy to wyłącznie od konkretnego rozwiązania, które przyjął programista. Najczęściej jednak poddawany jest kontroli realny czas jaki pozostał do ukończenia gry. Problem czasu w grach również nie jest zagadnieniem łatwym i generalnie rzecz biorąc stosują się do niego te same zasady postępowania, o jakich była mowa przy opisie energii.

Efekt obrazujący upływ czasu jest przedstawiany na ekranie najczęściej w postaci cyfrowej. Daje nam to dodatkową szansę uniknięcia mrówczej pracy związanej z wyłączaniem podprogramów. Szansą tą jest rozkaz SED (Set Decimal Mode), zmieniający tryb pracy mikroprocesora na dziesiętny, gdyż właśnie w tym trybie pracuje większość liczników „czasu" (zrozumiałe, wszak przed ekranem siedzi człowiek przyzwyczajony do systemu dziesiętnego, a nie binarnego).

```
338A RTS        ;koniec poprzedniego podprogramu
338B STX $23    ;rejestr X chwilowo w komórce $23
338D STY $32    ;rajestr Y chwilowo w komórce $32
339D SED        ;dziesiętny tryb pracy procesora
339E LDA $2F
33A0 ADC $2F    ;zawartość komórek $2F, $30, $31
33A2 STA $2F    ;jest podwajana i po tej operacji
33A4 LDA $30    ;przechowywana znów, odpowiednio
```

```
33A6 ADC $30  ;w komórkach $2F, $30, $31
33A8 STA $30  ;konstrukcje tego rodzaju są typowe dla
33AA LDA $31  ;liczników czasu, llosc komórek poddawana
33AC ADC $31  ;operacjom j/w jest zawsze o 1 mniejsza
33AF STA $31  ;ilości pozycji licznika wyświetlanych na
              ekranie
33B0 CLD      ;powrót do binarnego trybu pracy
33E1 RTS      ;koniec podprogramu
```

Tak więc bardzo często w pobliżu fragmentu programu kontrolującego „czas" figuruje rozkaz SED reprezentowany przez bajt o wartości #$F8. Dla „włamywacza" jest to informacja o kapitalnym znaczeniu. Spróbujmy w oparciu o nią zatrzymać liczniki wody i pożywienia w grze Chimera. Poszukiwanie rozkazu SED, podobnie jak SEI, jest nieco kłopotliwe. Najpraktyczniej jest przeglądać po 4 KB pamięci, a więc obszar $0000 – $1000, $1000––$2000 itd. pod kątem obecności bajtu #$F8. W przedziale $3000–$4000 monitor wyszukuje adres $339D. Po sprawdzeniu, że nie jest on fragmentem innego rozkazu, na ekranie widać następujące linie:

Zauważamy, że nasz „mechanizm zegarowy" znajduje się wewnątrz podprogramu zaczynającego się od adresu $338B. Sprawdźmy co się stanie, gdy w adres ten wpiszemy RTS. A więc: A338B RTS + RETURN i poprzez X + RETURN przechodzimy do Basic-a, aby po znalezionym wcześniej SYS $0A00 uruchomić grę. Na ekranie widać efekt naszej pracy, ale nie taki, jakbyśmy chcieli. Trzeba więc wrócić do programu, przywrócić poprzedni rozkaz, który był umieszczony od adresu $33B i sprawdzić, czy i skąd jest wywoływany JSR $338B. Pojawiają się adresy: $3410, $3447, $3C03. *Dis-asemblujemy* program i widzimy:

```
3400 LDA $12                   3437 LDA $14
3402 SBC #$01                  3439 SBC $02
3404 STA $12                   343B STA $14
3406 TAY                       343D TAY
3407 LDA $13                   343E LDA $15
3409 SBC #$00                  3440 SBC #$00
340B STA $13                   3442 STA $15
340D TAX                       3444 TAX
3624 LDA $12                   3B80 LDA $14
3626 SBC $02                   3B82 ADC #$DC
3628 STA $12   ;licznik wody   3B84 STA $14   ;licznik
362A LDA $13                   3B86 LDA $15   ;żywności
362C SBC $03                   3B88 ADC #$05
362E STA $12                   3B8A STA $15
3B97 LDA $12                   3C80 LDA $14
3B99 ADC #$DC                  3C82 ADC #$04
3B9B STA $12                   3C84 STA $14
3B9D LDA $13                   3C86 LDA $15
3B9F ADC #$05                  3C88 ADC #$09
3BA1 STA $13                   3C8A STA $15
```

Nie trzeba posiadać zbyt wielkiego doświadczenia w pracy z asemblerem aby stwierdzić, że tym razem znaleźliśmy to, o co nam chodziło. Zawartość komórek $12, $13 określa stan licznika wody, a $14, $15 — stan licznika pożywienia. Szczegółową analizę fragmentów programu j/w uważam za zbędną, warto jedynie zapamiętać charakterystyczne sekwencje rozkazów LDA, SBC, STA oraz LDA, ADC, STA jako typowe sposoby zmniejszania lub zwiększania zawartości komórek pamięci. Aby zatrzymać oba liczniki można wszystkie rozkazy SBC i ADC zastąpić rozkazami NOP, lub też w sposób bardziej elegancki, zmieniając wszystkie rozkazy STA na LDA, co wymaga zmiany jedynie jednego bajtu przy wprowadzaniu poprawek z Basic-a.

(cdn.)

*URAN*

# C-64 TROCHĘ „MĄDRZEJSZY"...

Jak zapewne każdy z Was wie, pisanie programów jest przyjemnością, a ich uruchamianie — zmorą. Zwłaszcza denerwujące jest, gdy „krnąbrny" komputer „nie chce" wykonać długiego programu w całości (pomimo, że jego koncepcja jest poprawna), ponieważ w jakiejś tam linii wpisaliśmy zamiast zera literę O lub popełniliśmy podobny, mało ważny błąd.

*Poniższy program przeznaczony dla C-64 (listing 1) powoduje, iż w przypadku wystąpienia błędu komputer w górnej części ekranu wyświetla w ramce (podobnej jak GURU na komputerze AMIGA) odpowiedni komunikat o błędzie oraz informację „(I) IGNORE IT (A) ABORT". Po wciśnięciu klawisza „I" zostaje przywrócony ekran sprzed komunikatu błędu i wykonywanie programu jest kontynuowane.* W pewnych przypadkach wybranie opcji IGNORE może spowodować, iż program zostanie wykonany niepoprawnie, co jest chyba oczywiste. Natomiast wciśnięcie klawisza „A" spowoduje powrót do BASIC-a (wyświetlenie komunikatu READY).

Dla wszystkich, którzy chcieliby zmodyfikować program, przedstawiam także listing (2) nieco okrojonego programu w asemblerze (Turbo Assembler 5.0) wraz z komentarzem.

*MARIUSZ FERDYN*

***Literatura:***

[1] Komputer 1/87, Programowa obsługa błędów w C-64, Roland Wacławek
[2] Jak rozbudować interpreter, Krzysztof Gajewski, Bogusław Radziszewski, SOETO 1989

LISTING 1

```
200 rem ***************************
205 rem *      super errors       *
210 rem *           by            *
215 rem *        m.ferdyn         *
220 rem ***************************
225 d=49152:b=62894
230 c=0:e=d
235 read a$:if a$="end" then 285
240 a1=asc (left$ (a$,1)) and 63
245 a2=asc (right$ (a$,1)) and 63
250 if a1>47 then 260
255 a1=a1+9:goto 265
260 a1=a1-48
265 if a2>47 then a2=a2-48:goto 275
270 a2=a2+9
275 a=a1*16+a2:poke d,a
280 d=d+1:c=c+a:goto 235
285 if c<>b then print "blad w liniach d
    ata":stop
290 print:print "activ. !!!"
295 sys e:new
300 data a2,0b,a9,c0,8e,00,03,8d
305 data 01,03,60,8a,10,03,4c,74
310 data a4,a4,39,8c,ea,c1,a4,3a
315 data 8c,eb,c1,8d,05,c2,a5,d3
320 data 8d,06,c2,a5,d6,8d,07,c2
325 data ad,86,02,8d,08,c2,ad,21
330 data d0,29,0f,c9,01,f0,08,a9
335 data 01,8d,86,02,4c,44,c0,a9
340 data 00,8d,86,02,a2,00,bd,00
345 data d8,9d,0e,c2,bd,00,d9,9d
350 data 0e,c7,bd,00,da,9d,0e,c8
355 data bd,00,db,9d,0e,c9,e8,d0
360 data e5,a2,00,bd,00,04,9d,0e
365 data c3,bd,00,05,9d,0e,c4,bd
370 data 00,06,9d,0e,c5,bd,00,07
375 data 9d,0e,c6,e8,d0,e5,a9,93
380 data 20,d2,ff,20,af,c1,a2,a0
385 data a9,20,9d,00,04,ca,d0,fa
390 data a9,13,20,d2,ff,a2,28,a9
395 data b9,20,d2,ff,ca,d0,fa,ad
400 data 05,c2,0a,aa,bd,26,a3,85
405 data 22,bd,27,a3,85,23,20,cc
410 data ff,a9,00,85,13,20,d7,aa
415 data 20,45,ab,a0,00,b1,22,48
420 data 29,7f,20,47,ab,c8,68,10
425 data f4,a9,69,a0,a3,20,1e,ab
430 data a4,3a,c8,f0,03,20,c2,bd
435 data a9,0d,20,d2,ff,a9,ec,a0
440 data c1,20,1e,ab,a9,0d,20,d2
445 data ff,a2,28,a9,b9,20,d2,ff
450 data ca,d0,fa,ad,00,04,8d,0c
455 data c2,ad,a0,04,8d,0d,c2,ee
460 data 09,c2,ad,09,c2,c9,ff,f0
465 data 03,4c,24,c1,ee,09,c2,ee
470 data 0b,c2,ad,0b,c2,c9,30,f0
475 data 45,4c,24,c1,a9,00,8d,09
480 data c2,8d,0b,c2,20,e4,ff,c9
485 data 41,f0,07,c9,49,f0,1f,4c
490 data ff,c0,a9,20,20,d2,ff,20
495 data 9d,c1,a9,0d,20,d2,ff,a9
500 data 76,a0,a3,20,1e,ab,a9,80
505 data 20,90,ff,6c,02,03,a9,20
510 data 20,d2,ff,a9,0d,20,d2,ff
515 data 20,9d,c1,4c,ae,a7,ad,0a
520 data c2,c9,02,f0,1a,a9,20,a2
525 data 28,9d,ff,03,ca,d0,fa,a2
530 data 28,9d,9f,04,ca,d0,fa,a9
535 data 02,8d,0a,c2,4c,1c,c1,ad
540 data 0c,c2,a2,28,9d,ff,03,ca
545 data d0,fa,ad,0d,c2,a2,28,9d
550 data 9f,04,ca,d0,fa,a9,01,8d
555 data 0a,c2,4c,1c,c1,ae,07,c2
560 data ac,06,c2,18,20,f0,ff,ad
565 data 08,c2,8d,86,02,a2,00,a2
570 data 00,bd,0e,c3,9d,00,04,bd
575 data 0e,c4,9d,00,05,bd,0e,c5
580 data 9d,00,06,bd,0e,c6,9d,00
585 data 07,e8,d0,e5,a2,00,bd,0e
590 data c2,9d,00,d8,bd,0e,c7,9d
595 data 00,d9,bd,0e,c8,9d,00,da
600 data bd,0e,c9,9d,00,db,e8,d0
605 data e5,60,00,00,28,49,29,20
610 data 49,47,4e,4f,52,45,20,49
615 data 54,20,20,28,41,29,20,41
620 data 42,4f,52,54,00,00,00,00
625 data 00,00,00,00,00,00,end
```

LISTING 2

```
        JSR $AAD7    ; WYPROWADZENIE CR I
        JSR $AB45    ; LF (POWRÓT KARETKI
                     ; I KONIEC WIERSZA
                     ; ORAZ "?"
        LDY #$00
LOOP    LDA ($22),Y
        PHA          ; TEKST KOMUNIKATU
        AND #$7F
        JSR $AB47    ; WYPROWADZENIE
        INY
        PLA
        BPL LOOP
        LDA #$69
        LDY #$A3
        JSR $AB1E    ; WYDRUK ERROR
        LDY $3A
        INY          ; TRYB BEZPOŚREDNI?
        BE\20\1 BEZP; TAK!
        JSR $BDC2    ; WYDRUK IN + NR
                     ; LINII
BEZP    LDA #$0D
        JSR $FFD2
        LDA #<KOM
        LDY #>KOM
        JSR $AB1E    ; WYDRUK KOMUNIKATU
        LDA #$00     ; ZEZWOLENIE NA
        STA $CC      ; PULSOWANIE KURSORA
KEY     JSR $FFE4
        CMP #$41
        BE\20\1 KEYA; WCIŚNIĘTO KLAWISZ
                     ; "A" (ABORT)
        CMP #$49
        BE\20\1 KEYI; WCIŚNIĘTO KLAWISZ
        *= $C000
;-----------------------------------------
        LDX #<START ; WSTAWIENIE NOWEGO
        LDA #>START ; WEKTORA PROCEDURY
        STX 768      ; OBSŁUGI BŁĘDÓW
        STA 769
        RTS
;-----------------------------------------
START   TXA          ; NR BŁĘDU DO AKUM.
        BPL BLAD
        JMP 42100    ; BEZ BŁĘDU
BLAD    LDY 57
        STY NRBM     ; ZAPAMIĘTANIE NR-U
        LDY 58       ; LINII, W KTÓREJ
        STY NRBS     ; WYSTAPIŁ BŁĄD
        ASL A
        TAX
        LDA $A326,X ; ADRES KOMUNIKATU
        STA $22
        LDA $A327,X
        STA $23
        JSR $FFCC
        LDA #$00     ; FLAGA WE/WY
        STA $13
                     ; "I" (IGNORE IT)
        JMP KEY
KEYA
        LDA #$20
        JSR $FFD2
        LDA #$01     ; ZABLOKOWANIE
        STA $CC      ; PULSOWANIA KURSORA
        LDA #$76
        LDY #$A3
        JSR $AB1E    ; WYDRUK READY
        LDA #$80
        JSR $FF90    ; WŁĄCZENIE TRYBU
                     ; BEZPOŚREDNIEGO
        JMP ($0302) ; SKOK DO GŁÓWNEJ
                     ; PĘTLI PROGRAMOWEJ
                     ; INTERPRETERA BASIC
KEYI
        LDA #$20
        JSR $FFD2
        LDA #$0D
        JSR $FFD2
        LDA #$01     ; ZABLOKOWANIE
        STA $CC      ; PULSOWANIA KURSORA
        JMP $A7AE    ; SKOK DO PĘTLI
                     ; INTERPRETERA
;-----------------------------------------
NRBM    .BYTE 0
NRBS    .BYTE 0
KOM     .TEXT "(A) ABORT  (I) IGNORE IT"
        .TEXT " "
        .BYTE 0
```

# DLA KAŻDEGO COŚ MIŁEGO (2)

***Spotykamy się już po raz drugi (i mam nadzieję, że nie ostatni) w naszej rubryce „Dla każdego coś miłego". Przedstawiamy w niej programy zwariowane i takie sobie, przydatne lub nieprzydatne, słowem takie, które mogą czymś zaciekawić. W redagowaniu tej rubryki możecie mieć udział także i Wy. Czekamy zatem na ciekawe pomysły i programy od Was.***

Dziś kolejna porcyjka świeżych (chrupiących) programików prosto z klawiatury. Jako pierwszy proponujemy program o nazwie BORDER.

Zapewne widziałeś już nieraz programy demonstracyjne, w których w miejscu ramki (border) przesuwał się duszek. Na pewno zastanawiałeś się, jak to się robi. Przestudiuj listing 1 i 2, a problem ten przestanie być dla Ciebie tajemnicą.

Program BORDER działa na przerwaniach rastra. Po jego uruchomieniu zniknie dolna i górna ramka. Następnie należy uruchomić program z danymi dla duszka (listing 2)... i już!

Pamiętacie dwa programy antyreset z poprzedniego odcinka? Oto trzeci, jeszcze inny program o tym profilu (patrz listing 3).

W przypadku wciśnięcia kombinacji klawiszy RUN/STOP-RESTORE, samego RESTORE lub klawisza reset komputer zignoruje te działanie i uruchomi program rezydujący w pamięci komputera (w języku BASIC). Antyreset #3 nie będzie jednak działał w przypadku użycia przycisku reset w modułach FINAL II, III lub Action Replay.

Jak zrobić płynny *scrolling?* To pytanie nurtuje wielu z Was. Programy napisane w języku Basic działają zbyt wolno i w dodatku tekst nie jest przesuwany płynnie. W związku z tym trzeba uciec się do języka maszynowego. Program z listingu 4 służy właśnie do produkcji takiego płynnie przesuwającego się tekstu.

Jego obsługa jest prosta. Tekst, który ma występować w *scrollu* wpisujemy w liniach DATA (tak jak zostało to zrobione na przykładzie). Dodatkowo można regulować szybkość przesuwu tekstu przy pomocy instrukcji POKE:

**POKE 49204,121** — wolno,
**POKE 49204,127** — średnio,
**POKE 49204,139** — szybko,
**SYS 49152** — uruchomienie *scrollu*

Na koniec mały i zapewne znany wielu użytkownikom C-64 przydatny trick.

Zdarzyło Ci się zapewne kopiować program na kasetę lub dyskietkę. Po wpisaniu przepisowego SAVE i tytułu programu oraz naciśnięciu klawisza RETURN na ekranie zamiast upragnionego komunikatu SAVING pojawił się napis: OUT OF MEMORY ERROR. Po wielu krwawych walkach z programem doszedłeś do wniosku, że możliwy jest jego zapis, lecz niestety bez tytułu. W przypadku nagrywania na taśmę nie ma to większego znaczenia lecz przy zastosowaniu dyskietki problem urasta do strasznych rozmiarów. Odczyt z dyskietki pliku nie posiadającego nazwy jest prawie niemożliwy.

Co zatem można zrobić? Ano wystarczy wpisać **POKE 56,208** po ukazaniu się komunikatu o braku pamięci i ponownie wykonać procedurę zapisu. W większości wypadków program zapisze się na taśmie lub dyskietce z tytułem jaki mu nadaliśmy.

*MARIUSZ FERDYN*
*PIOTR LISZEWSKI*

```
1 REM * LISTING 1 *
2 :
10 D=49152:B=11349
15 C=0:E=D
20 READ A$:IF A$="END" THEN 70
25 A1=ASC(LEFT$(A$,1))AND 63
30 A2=ASC(RIGHT$(A$,1))AND 63
35 IF A1>47 THEN 45
40 A1=A1+9:GOTO 50
45 A1=A1-48
50 IF A2>47 THEN A2=A2-48:GOTO
   60
55 A2=A2+9
60 A=A1*16+A2:POKE D,A
65 D=D+1:C=C+A:GOTO 20
70 IF C<>B THEN PRINT "BLAD W L
   INIACH DATA":STOP
71 POKE 16383,0:REM WYZEROWANIE
   OSTATNIEGO BAJTU W BANKU VIC
75 SYS E
100 DATA 78,A9,01,8D,1A,D0,A9,7F
101 DATA 8D,0D,DC,A2,21,A0,C0,8E
102 DATA 14,03,8C,15,03,A9,1B,8D
103 DATA 11,D0,A9,F9,8D,12,D0,58
104 DATA 60,EE,19,D0,A9,16,AE,12
105 DATA D0,EC,12,D0,F0,FB,8D,11
106 DATA D0,A9,3E,8D,14,03,A9,28
107 DATA 8D,12,D0,4C,31,EA,EE,19
108 DATA D0,A9,1B,AE,12,D0,EC,12
109 DATA D0,F0,FB,8D,11,D0,A9,21
110 DATA 8D,14,03,A9,F9,8D,12,D0
111 DATA 4C,31,EA,END
```

```
100 REM * LISTING 2 *
101 :
200 FOR X=704 TO 704+62:READ A:P
    OKE X,A:NEXT:REM WZOR
210 POKE 2040,11:REM WSKAZNIK NA
    OBSZAR OD ADRESU 704
220 POKE 53287,1:REM KOLOR SPRIT
    E'A BIALY
230 POKE 53248,24:REM WSPOLRZEDN
    A X
240 POKE 53249,255:REM WSPOLRZED
    NA Y
250 POKE 53269,1:REM WLACZENIE S
    PRITE'A
260 REM SIR ALEC
270 DATA 001,255,128,015,255,240
    ,063,255
280 DATA 252,063,255,252,127,255
    ,254,127
290 DATA 255,254,120,126,030,111
    ,153,246
300 DATA 120,126,030,112,060,014
    ,048,060
310 DATA 012,024,126,024,015,231
    ,240,011
320 DATA 195,208,007,153,224,007
    ,255,224
330 DATA 002,211,064,002,211,064
    ,001,255
340 DATA 128,000,231,000,000,000
    ,000
```

```
100 REM * LISTING 3 *
101 :
200 DATA 009,128,009,128,195,194
201 DATA 205,056,048,120,032,163
202 DATA 253,032,080,253,032,021
210 DATA 253,032,091,255,032,083
211 DATA 228,032,191,227,032,034
212 DATA 228,169,001,141,002,008
220 DATA 032,051,165,165,034,133
221 DATA 045,165,035,133,046,032
230 DATA 094,166,032,085,166,076
231 DATA 174,167,032
232 :
240 FOR J=32768 TO 32824:READ Q:
    POKE J,Q:NEXT
250 PRINT "PROGRAM AKTYWNY.":NEW
```

```
100 REM * LISTING 4 *
101 :
200 D=49152:B=17831
210 C=0:E=D
220 READ A$:IF A$="END" THEN 320
230 A1=ASC(LEFT$(A$,1))AND 63
240 A2=ASC(RIGHT$(A$,1))AND 63
250 IF A1>47 THEN 270
260 A1=A1+9:GOTO 280
270 A1=A1-48
280 IF A2>47 THEN A2=A2-48:GOTO
    300
290 A2=A2+9
300 A=A1*16+A2:POKE D,A
310 D=D+1:C=C+A:GOTO 220
320 IF C<>B THEN PRINT "BLAD W L
    INIACH DATA 1":STOP
321 :
330 DATA 78,AE,14,03,AC,15,03,8E
340 DATA 77,C0,8C,78,C0,A0,C0,A2
350 DATA 2D,8E,14,03,8C,15,03,A9
360 DATA 7F,8D,0D,DC,A9,01,8D,1A
370 DATA D0,A9,1B,8D,11,D0,A9,F1
380 DATA 8D,12,D0,58,60,AD,16,D0
390 DATA 8D,52,C0,20,79,C0,A9,39
400 DATA 8D,12,D0,A9,51,8D,14,03
410 DATA AD,11,D0,29,7F,8D,11,D0
420 DATA AD,19,D0,8D,19,D0,4C,81
430 DATA EA,A9,C8,AC,12,D0,CC,12
440 DATA D0,F0,FB,8D,16,D0,A9,00
450 DATA 8D,12,D0,A9,2D,8D,14,03
460 DATA AD,11,D0,29,7F,8D,11,D0
470 DATA AD,19,D0,8D,19,D0,4C,31
480 DATA EA,20,0A,C1,4C,0A,C1,20
490 DATA 0A,C1,20,0A,C1,20,0A,C1
500 DATA 4C,0A,C1,20,0A,C1,20,0A
510 DATA C1,20,0A,C1,20,0A,C1,20
520 DATA 0A,C1,20,0A,C1,20,0A,C1
530 DATA 4C,0A,C1,END
540 D=49408:B=12069
550 C=0:E=D
560 READ A$:IF A$="END" THEN 660
570 A1=ASC(LEFT$(A$,1))AND 63
580 A2=ASC(RIGHT$(A$,1))AND 63
590 IF A1>47 THEN 610
600 A1=A1+9:GOTO 620
610 A1=A1-48
620 IF A2>47 THEN A2=A2-48:GOTO
    640
630 A2=A2+9
640 A=A1*16+A2:POKE D,A
650 D=D+1:C=C+A:GOTO 560
660 IF C<>B THEN PRINT "BLAD W L
    INIACH DATA 2":STOP
661 :
670 DATA A2,00,A0,C2,8E,0B,C1,8C
680 DATA 0C,C1,AD,00,C2,C9,00,F0
690 DATA EF,8D,27,04,A9,C0,8D,16
700 DATA D0,AE,15,C1,E0,C0,F0,09
710 DATA CA,8E,15,C1,60,EA,EA,EA
720 DATA EA,A9,C0,8D,16,D0,A2,00
730 DATA BD,01,04,9D,00,04,E8,E0
740 DATA 28,D0,F5,A9,C7,8D,16,D0
750 DATA A9,C7,8D,15,C1,AD,0B,C1
760 DATA C9,FF,F0,04,EE,0B,C1,60
770 DATA EE,0B,C1,EE,0C,C1,60,END
780 DIM TEKST$(10)
790 I=0
800 READ TEKST$(I)
810 IF TEKST$(I)="@" THEN 840
820 I=I+1
830 GOTO 800
840 W=49664
850 K=PEEK(1024)
860 FOR R=0 TO I
870 D=LEN(TEKST$(R))
880 FOR A=1 TO D
890 A$=MID$(TEKST$(R),A,1)
900 PRINT CHR$(147);A$
910 POKE W,PEEK(1024)
920 W=W+1
930 NEXT A
940 NEXT R
950 POKE 1024,K
960 CLR:REM SKASOWANIE WSZYSTKIC
    H ZMIENNYCH WYSTEPUJACYCH W
    PROGRAMIE
970 REM POKE 49204,121 - WOLNO
980 REM POKE 49204,127 - SZYBKA
990 REM POKE 49204,139 - DLA KIL
    LEROW
1000 REM SYS 49152 -URUCHOMIENIE
1010 POKE 49204,121:REM WOLNA
1020 SYS 49152:REM URUCHOMIENIE
1030 END
1040 DATA " MARSOFT AND PLIS SEN
     D HOT GREETS TO ALL COMMODO
     RE COMPUTERS"
1050 DATA " USERS AND ALL COMPUT
     ER FREAKS IN POLAND. HI BUD
     DIES..... "
1060 DATA " BEST WISHES FROM C&A
     STAFF TO ALL COMMODORE TEAM
     S IN"
1070 DATA "POLAND. SEE YOU SOON
     ON A VERY SPECIAL MEETING !
     .......             "
1080 DATA @
```

# KALKULATOR

Program ten służy do przeliczenia liczby w systemie dziesiętnym na liczbę w dowolnym systemie liczbowym o podstawie nie większej niż 16. Obliczona liczba może mieć maksymalnie 8 miejsc.

Program działa według następującego algorytmu:

1. Obliczamy tablice potęg liczby „P" wprowadzonej jako podstawa systemu (linie programu 210 do 240).
2. Dzielimy wprowadzoną liczbę „L" przez potęgę „H(7)" liczby „P", odrzucamy część ułamkową i wynik zapamiętujemy w tabeli jako „D(O)".
3. Obliczamy resztę „C(O)" C(O)=L-[H(7)*D(O)].
4. Sprawdzamy czy D(O)>9 (linia programu 530).
   4.1 Jeśli jest to prawda, to „D(O)" zamieniamy na kolejne litery t.j.: A=10, B=11, C=12...itd. =E$(N) (linie programu 430 do 520).
5. Dzielimy resztę „C(O)" przez „H(6)".
6. Obliczamy resztę „C(1)":C(1)=C(O)-[H(6)*D(1)].
7. Powtarzamy operacje 4 do 6 zmniejszając indeks „H(N)".
8. D(9)=C(6).
9. Jeżeli D(N)>9 to D(N)=E$(N).
10. Zapisujemy otrzymaną liczbę w kolejności D(O) do D(7).

*WOJCIECH BENBENEK*

*OD REDAKCJI:*
*Program nie jest zbyt ciekawy graficznie (nawiasem mówiąc autorowi bynajmniej na tym nie zależało), natomiast może być niezastąpiony jako pomoc naukowa do wkuwania matematyki. Kto chce, może sobie oczywiście samemu dorobić lepszą szatę graficzną, tu zostawiamy już Czytelnikom pole do popisu.*

```
10 rem ***********************
20 rem *   'kalkulator'      *
60 rem *       autor         *
70 rem * wojciech benbenek *
80 rem ***********************
140 :
150 printchr$(144)
160 poke53280,11:poke53281,15
170 printchr$(147)
200 input"wprowadz podstawe systemu 2>=p
    <=16";p
210 y=0
220 for f=0 to 8
230 h(f)=int(p^y)
240 y=y+1:next
250 a$(1)=" a":a$(2)=" b":a$(3)=" c":a$(
    4)=" d":a$(5)=" e":a$(6)=" f"
260 a$(8)=" h":a$(9)=" j":printchr$(147)
270 y=0
280 forf=0to7
290 printp;"^";y;"=";int(h(f)):y=y+1
300 next:print:print
310 print"wpisz liczbe '=<";int(h(8)-2);
    "' ";:input  l$:l=val(l$)
320 gosub550
330 print:print;chr$(28);l,chr$(144)"dzi
    esietnie ="
340 print:print:printspc(3)
350 forx=0tc7:printchr$(31)e$(x);:next:p
    rint,chr$(156);p;chr$(144)"-owo"
390 print:print:print:print:print:print"
    jeszcze cos liczymy (t/n)"
400 getx$:ifx$=""then400
410 ifx$="t"then170
460 printchr$(147):end
470 e$(0)=str$(d(0)):n=d(0)-9:e$(0)=a$(n
    ):return
480 e$(1)=str$(d(1)):n=d(1)-9:e$(1)=a$(n
    ):return
490 e$(2)=str$(d(2)):n=d(2)-9:e$(2)=a$(n
    ):return
500 e$(3)=str$(d(3)):n=d(3)-9:e$(3)=a$(n
    ):return
510 e$(4)=str$(d(4)):n=d(4)-9:e$(4)=a$(n
    ):return
520 e$(5)=str$(d(5)):n=d(5)-9:e$(5)=a$(n
    ):return
530 e$(6)=str$(d(6)):n=d(6)-9:e$(6)=a$(n
    ):return
540 s$(7)=str$(d(7)):n=d(7)-9:e$(7)=a$(n
    ):return
550 d(0)=int(l/h(7)):c(0)=l-(h(7)*d(0)):
    ifd(0)>9thengosub470
560 ifd(0)<10thene$(0)=str$(d(0))
570 d(1)=int(c(0)/h(6)):c(1)=c(0)-(h(6)*
    d(1)):ifd(1)>9thengosub480
580 ifd(1)<10thene$(1)=str$(d(1))
590 d(2)=int(c(1)/h(5)):c(2)=c(1)-(h(5)*
    d(2)):ifd(2)>9thengosub490
600 ifd(2)<10thene$(2)=str$(d(2))
610 d(3)=int(c(2)/h(4)):c(3)=c(2)-(h(4)*
    d(3)):ifd(3)>9thengosub500
620 ifd(3)<10thene$(3)=str$(d(3))
630 d(4)=int(c(3)/h(3)):c(4)=c(3)-(h(3)*
    d(4)):ifd(4)>9thengosub510
640 ifd(4)<10thene$(4)=str$(d(4))
650 d(5)=int(c(4)/h(2)):c(5)=c(4)-(h(2)*
    d(5)):ifd(5)>9thengcsub520
660 ifd(5)<10thene$(5)=str$(d(5))
670 d(6)=int(c(5)/h(1)):c(6)=c(5)-(h(1)*
    d(6)):ifd(6)>9thengosub530
680 ifd(6)<10thene$(6)=str$(d(6))
690 d(7)=int(c(6)):ifd(7)>9thengosub540
700 ifd(7)<10thene$(7)=str$(d(7))
710 return
```





# KOMÓRKA $ D016 – NIEOGRANICZONE MOŻLIWOŚCI?

Przerwania rastra w C-64 to naprawdę potężne narzędzie, umożliwiające pomysłowemu programiście uzyskanie efektów, które wprawią potencjalnego użytkownika komputera w ogromne zdziwienie. Chciałbym przedstawić jeden z prostszych i ciekawszych trików, mianowicie efekt falowania napisów czy logosów. Zapewne każdy z Czytelników spotkał się już z czymś takim w demach lub intrach. Jak przystało na dobry trik, jest on bardzo prosty. Wiadomo, że komórka $D016 służy m.in. do przesuwania zawartości ekranu w osiem punktów. A co by było, gdyby zmieniać zawartość tej komórki co każdą linię ekranu? Prosty i stateczny rysunek można w ten sposób trochę „powyginać". Jeśli utworzymy tablicę (najlepiej obliczoną funkcją sinusoidalną) zawierającą kolejne wartości wpisywane do $D016, to nasz rysunek wygnie się w ładną falę. Dla spotęgowania efektu można po każdym przerwaniu przesunąć zawartość tablicy, co spowoduje ruch fali.

Jako przykład przedstawiam poniższy program. Bardziej wnikliwym polecam przestudiowanie kodu maszynowego.

*ROBERT KUNA*

```
100 for a=0 to 31:read d:poke 49408+a,d:
    next
105 a=49152
110 read d:if d=999 then 120
115 poke a,d:a=a+1:goto 110
120 poke 53280,0:poke 53281,0:poke646,15
125 printchr$(147)"(12 spacji)Commodore"
130 print"(16 spacji)64"
135 sys 49152:end
140 data 0,0,1,1,1,2,2,3,3,4,4,5,5,5,6,6
145 data 6,6,5,5,5,4,4,3,3,2,2,1,1,1,0,0
150 data 120,169,127,141,013,220,162,000
155 data 142,014,220,232,142,026,208,169
160 data 027,141,017,208,169,049,141,018
165 data 208,169,037,141,020,003,169,192
170 data 141,021,003,088,096,160,000,174
175 data 018,208,232,236,018,208,208,251
180 data 185,000,193,141,022,208,200,192
185 data 016,208,236,169,200,141,022,208
190 data 173,000,193,072,169,000,189,001
195 data 193,157,000,193,232,224,032,208
200 data 245,104,141,031,193,238,025,208
205 data 076,049,234,999
```

# FRAKTAL

Program zamieszczony poniżej jest przeznaczony dla C-128, ale bez trudu powinien dać się przerobić tak, aby działał na C-16/116 i +4. Prezentuje on trochę inne niż zwykle podejście do fraktali, a mianowicie rysuje krzywą Mandelbrota. Zmieniając trzy parametry, o które program pyta na początku, możemy rysować krzywe różnego stopnia. Np. wybierając odpowiednio zakres można oglądać różne fragmenty tego samego fraktala i badać samopodobieństwo (najlepiej, gdy parametry są z zakresu -2 do +2). Podczas rysowania wciśnięcie klawisza „space” powoduje przełączenie zegara komputera na wyższą częstotliwość (tryb FAST), co prawda obraz wtedy zniknie, ale prędkość obliczeń wzrośnie prawie dwukrotnie. Naciśnięcie klawisza „return” przywraca standardową prędkość i pokazuje, ile komputer do tej pory policzył.

Życzę udanej zabawy i wiele cierpliwości. Warto jest skompilować program, gdyż przyspieszy to kilkakrotnie proces rysowania.

*RAFAŁ PIASEK*

```
100 rem ********************************
105 rem * program by jetboy/parados *
110 rem * (c)commodore & amiga 1992 *
115 rem ********************************
120 rem
125 graphic0,1
130 color1,16:color2,13:color3,12
135 poke53280,0:poke53281,0:scnclr
140 input"po ilu iteracjach zmieniac ko
    lor1";k1
145 input"po ilu iteracjach zmieniac ko
    lor2";k2
150 if k2<=k1thenprint"kolor2 musi byc
    wiekszy niz kolor1":print:goto145
155 input"po ilu iteracjach zmieniac ko
    lor3";k3
160 if k3<=k2thenprint"kolor3 musi byc
    wiekszy niz kolor2":print:goto155
165 print:print:print"okresl zakres"
170 print:input"podaj x minimalne";mx
175 input"podaj x maksymalne";wx
180 if mx>=wx then goto 175
185 print:input"podaj y minimalne";my
190 input"podaj y maksymalne";wy
195 if my>=wy then goto 190
200 input"wszystko ok (t/n)";a$
205 if a$="t" then 215
210 run
215 rem ----------------------------
220 graphic3,1
225 sx=(wx-mx)/100
230 sy=(wy-my)/200
235 yp=my-sy
240 for y=0 to 200
245 yp=yp+sy
250 xp=mx-sx
255 for x=0 to 100
260 xp=xp+sx
265 rs=0:us=0:n=0
270 rn=(re*rs-us*us)+xp
275 un=(ue*rs*2)+yp
280 if rn*rn+un*un>=4 then goto 300
285 rs=rn:us=un:n=n+1
290 if n=k3 thrn goto 315
295 goto 270
300 if n<k1 then c=1:goto320
305 if n<k2 then c=2:goto320
310 if n<k3 then c=3:goto320
315 c=0
320 draw c,x,y
325 get a$:if a$=""then goto 340
330 if a$=" " then fast:goto 340
335 if a$=chr$(13) then slow
340 next x
345 next y
350 print"":slow
355 get a$:if a$="" then goto 355
360 run
```

Magnetofon — magiczne słowo spędzające sen z oczu wielu posiadaczom komputerów Commodore. Takie to małe, pozornie nieskomplikowane, a często dużo z tym kłopotów. Prawdą jest, że magnetofony to już przeżytek i korzystanie z nich jako z pamięci zewnętrznej jest traktowane jako relikt przeszłości. W Polsce jednak magentofon ciągle utrzymuje swoją pozycję i ma nawet zwolenników, głównie ze względu na cenę. Aby Twój magnetofon pracował długo i niezawodnie, staraj się stosować do poniżej opisanych zasad:

*1. Nigdy nie rozkręcaj i nie dłub w sprzęcie elektronicznym, jeśli nie masz pojęcia o elektronice. Bardzo łatwo jest coś zepsuć powodując zwarcie — dotyczy to zwłaszcza sprzętu komputerowego. W starszych modelach Commodore magnetofon jest przyłączany bezpośrednio do mikroprocesora i uszkodzenie magnetofonu powoduje zwykle zniszczenie tego układu, co wiąże się z dość wysokimi wydatkami.*

*2. Nie zalecam w żadnym wypadku kręcenia głowicą w celu dopasowania jej skosu do taśm serdecznego przyjaciela Ciupasa. Efektem ciągłych zmian skosu głowicy są programy ponagrywane każdy na innym poziomie oraz nadmierny luz w połączeniu otwór-wkręt. W tym ostatnim wypadku zawodność sprzętu znacznie wzrasta.*

*3. Aby uniknąć, w przypadku natręctwa kopiowania i niechęci do regulacji głowicy, ciągłego przekładania wtyczek magnetofonów przy włączonym komputerze, co jest BARDZO niebezpieczne dla samego komputera, należy wykonać lub zamówić interfejs pozwalający na podłączenie dwóch Datasette 1531 (schemat i sposób wykonania w BAJTKU nr 3/4 1986). Można także pożyczyć od kolegi stację dysków i przenieść oprogramowanie za jej pośrednictwem.*

# CZEGO MAGNETOFON NIE LUBI?

*4. Magnetofon nie lubi, gdy zdejmuje mu się kieszeń, aby łatwiej wkładać i wyjmować kasetę (potrafi ona sama wypaść, jeśli trzaskasz jej wieczkiem), nie znosi również brudu, pogiętych taśm, tłuszczu itp.*

*5. Nie dotykaj głowicy palcami ani czymkolwiek innym — może to spowodować jej zabrudzenie lub uszkodzenie.*

*6. Staraj się nie stosować kaset metalowych (chromowych i żelazowych), gdyż twartość nośnika nie będzie wpływała korzystnie na żywotność głowicy, a kolorowe pudełko i imponująca cena kasety nie spowodują polepszenia się zapisu. Najlepiej stosować kasety zwykłe i jednocześnie najtańsze (dawniej z powodzeniem stosowałem kasety z nagranymi bajkami i szmatławymi piosenkami).*

*7. Magnetofon lubi natomiast, żeby od czasu do czasu przeczyścić mu głowicę i gumową rolkę transportera (byle nie stosować wody z mydłem, rozpuszczalnika nitro czy spirytusu salicylowego). W sprzedaży znajdują się specjalne kasety czyszczące z płynem, a rolkę można wyczyścić denaturatem.*

*8. Raz do roku można poprosić kogoś, kto się na tym zna, o przeczyszczenie i konserwację mechanizmu magnetofonu w celu jego dalszej eksploatacji.*

*9. W nowych magnetofonach (kupowanych w sklepach PEWEX-u) przy wtyczce mająa się mały kabelek. Opracowany został jako uziemienie, jednakże ani razu nie widziałem komputera, do którego można by go podłączyć. Proponuję odciąć go i traktować jako kupkę złomu lub jako śmiertelnego wroga C-64 (wisząc przy wtyczce kabelki te lubią się wpychać do USER PORTU, co powoduje niezbyt przyjemny fajerwerk).*

*10. Magnetofony do Commodore przystosowane są do pracy w pozycji horyzontalnej (leżącej). Potrząsanie, podrzucanie, kopanie i uderzanie oraz ustawianie pionowo (na wzór magnetofonu w wieży) będzie wpływało ujemnie na jego pracę.*

Każdy kto będzie korzystał z powyższych rad nie powinien mieć żadnych problemów z magnetofonem Datasette 1531, Datacorder czy Phonemark, czyli ogólnie z magnetofonami współpracującymi z Commodore 64.

*PIOTR LISZEWSKI*

# CO OFERUJE EXPERT?

Chciałbym nieco przybliżyć Czytelnikom C&A ten stosunkowo stary już moduł, jednak momentami bijący na głowę ACTION REPLAY. Krótki opis EXPERTA odnajdziecie w artykule „SPEC na każdą okazję" (C&A nr 2/92), więc przejdę od razu do opisu plików zawartych na dysku sprzedawanym wraz z modułem.

Dysk zawiera dwa rodzaje plików: programy pomocnicze do EXPERTA, oraz jego „systemy operacyjne" — programy, które wczytuje się do pamięci RAM modułu — i dopiero wtedy można z nich korzystać. Oto lista tych plików:

## MAIN BACKUP v2.10.

Jest to niejako główny program wczytywany do pamięci RAM modułu. Z poziomu C-64 program ten (po zainstalowaniu w EXPERCIE) uruchamia się wciśnięciem klawisza RESTORE, po czym na ekranie pojawia się plansza tytułowa monitora. Poza standardowymi poleceniami monitora (takimi jak ASSEMBLE, HUNT, COMPARE itp.) program ten oferuje:

- wykonanie EOR na zadanym obszarze pamięci;
- zwiększenie komórek pamięci określonego obszaru pamięci o określoną wartość (OO–FF);
- rozkaz NEW (SYS 64738);
- PROTECT NEW — „zerowanie" pamięci w celu uniknięcia kompresji zbędnych danych;
- RESTART — powrót do „zamrażanego" programu, z uwzględnieniem wprowadzonych zmian;
- kompresję i zapis programu na dysku (program poddany kompresji w pamięci można ponownie uruchomić!);
- ustalanie adresu początkowego procedury rozpakowującej;
- listowanie programu w BASIC z poziomu monitora (!);
- sprawdzanie adresów początku i końca pliku znajdującego się na dysku;
- zmiana kolorów ekranu.

Procedura FREEZE wykorzystuje przerwania NMI. Mogą więc wystąpić problemy przy pracy z programami, które także korzystają z tych przerwań. Nie dziwcie się więc, jeśli wystąpi np. „przeskakiwanie" do EXPERTA — w takim przypadku wystarczy wykonać polecenie RESTART i program działa dalej. EXPERT wykorzystuje właściwie tylko 256 bajtów ($DCOO–$DCFF), co czyni go bardzo trudnym do wykrycia (są to rejestry CIA#1).

## NO BOOT BLOCK

Służy do instalacji programu przyspieszającego wczytywanie z dyskietki. Tworzony plik wynikowy ma długość 0 bloków (zapisany jest w wolnych sektorach katalogu dyskietki). BOOT TRILOGIC — bo tak nazywa się ten dopalacz — umożliwia wyświetlanie obrazków w formacie EXPERTA i wczytywanie programów z 6-krotnym przyspieszeniem. NO BOOT BLOCK pozwala też na szybkie formatowanie dysków (9,5 s).

## SPRITE EXTRACTOR v2.9sp.

Kolejny monitor, przeznaczony tylko i wyłącznie do obsługi sprite-ów.

## IMMORTALISER v2.9.

Jest to następny monitor o możliwościach zbliżonych do MAIN BACKUP v2.10, z tym, że dochodzą tu jeszcze polecenia do usuwania kolizji sprite-ów między sobą i z otoczeniem. Jest tu również jedno ciekawe polecenie, nie publikowane w instrukcji (.Y). Właściciele EXPERTA mogą przekonać się, że niektórym programistom czasami naprawdę się nudzi...

## SPRITES EDITOR

Jak sama nazwa wskazuje, jest to po prostu edytor sprite-ów.

## SCREEN GRABBER v2.9pi.

Ten monitor służy do „wyciągania" z programów rysunków w dowolnym formacie. Zapisuje je na dysku we własnym, dosyć specyficznym formacie.

## KOALA CONVERTER

Przydaje się do konwersji grafiki z formatu EXPERT HIRES na KOALA PAD i na odwrót.

## KOALA CRUNCHER

Program do kompresji grafiki w formacie KOALA PAD.

## KOALA FLIP

Umożliwia obrót obrazu w formacie KOALA PAD z „lewej" na „prawą".

## PRINTSHOP/KOALA

Służy do konwersji grafiki w formacie PRINTSHOP na KOALA PAD. Z powodu różnic w formatach (KOALA-LORES, PRINTSHOP-HIRES) niektóre detale rysunku mogą być utracone.

## DOODLE/KOALA

Konwersja DOODLE na KOALA PAD (HIRES na MILTICOLOR).

## KOALA ZOOM IN

Ten programik jest naprawdę godny uwagi — wczytuje z dyskietki obrazek w formacie DOODLE, PRINTSHOP, NEWSROOM, VIDCOM NORMAL, ART STUDIO, KOALA, VIDCOM MULTICOLOR, BLAZING, PADDLES lub GRAPHIC CONSTRUCTION SET i tworzy z wybranego wycinka obrazu OVERSCAN (obraz wyświetlany na całym ekranie, łącznie z ramką). Taki obrazek można zapisać na dyskietce jako plik, który po uruchomieniu wyświetla tak przetworzony obraz.

## DOODLE/PRINTSHOP

Konwersja DOODLE na PRINTSHOP.

## PRINTSHOP/DOODLE

Umożliwia konwersję grafiki z formatu PRINTSHOP na format DOODLE.

## PRINTSHOP FLIP

Odwraca obraz z „prawej" strony na „lewą" (w formacie PRINTSHOP).

## MULTI BACK v2.9m (EXPERT USERCLUB).

Monitor przeznaczony do „łamania" wieloczęściowych programów taśmowych. Po kompresji można je zapisać na dyskietce.

## DISK/TAPE v2.4.

Ten monitor służy do kopiowania programów z dysku na taśmę. EXPERT stosuje do zapisu na taśmie dosyć specyficzny format. Przypomina to trochę TSL, ale są to programy samouruchamiające się.

## BOOT

Jest to 6-krotnie przyspieszający dopalacz, instalowany przez program NO BOOT BLOCK. Nie zajmuje (teoretycznie) miejsca na dysku. Po wczytaniu i uruchomieniu najpierw wczytuje katalog dyskietki, następnie wyświetla go i czeka na wybór programu do wczytania.

## TAPE TO TAPE v2.9t.

Coś dla użytkowników magnetofonu. Zestaw poleceń taki sam jak w MAIN BACKUP v2.10, ale polecenia zapisu skompresowanych plików dotyczą taśmy. Zapis na taśmie w formacie przypominającym TSL (autorun).

Mimo, że EXPERT jest już dosyć starym wyrobem (ma prawie 4 lata), to jednak posiada wiele cech stawiających go znacznie wyżej niż FINAL III, czy ACTION REPLAY. Nawet jeżeli czegoś nam brakuje w monitorze EXPERTA, to zawsze można się pokusić o stworzenie własnego systemu operacyjnego.

Z mojego punktu widzenia EXPERT ma dwie wady: po pierwsze konieczność wczytywania programów do pamięci RAM karty (bez czego jednak niemożliwe byłoby osiągnięcie takiej uniwersalności) oraz fakt, że do pełnego wykorzystania tego modułu potrzebna jest solidna dawka wiedzy.

I to już właściwie wszystko na temat wyrobu firmy TRILOGIC. Autorami większej części systemu operacyjnego EXPERTA są J. Twiddy i G. Kelly — ludzie, którzy wykonali naprawdę niezły kawałek roboty.

Wersja, którą opisałem w tym artykule nosiła numer 2.9; wiem jednak, że istnieje już od dawna wersja 3.1. Jeżeli posiadacie ten moduł i wiecie coś ekstra na jego temat — napiszcie.

*BARTŁOMIEJ DRAMCZYK*

# FLI CO TO JEST I Z CZYM SIĘ TO JE

## COŚ DLA KONESERÓW

Jak zapewne wiecie, Commodore 64 posiada pewne ograniczenia w wyświetlaniu grafiki. W trybie *multicolor* (160x200 punktów) mamy do dyspozycji 16 kolorów. Niestety, w polu o wielkości jednego znaku (4x8 punktów), mamy do dyspozycji tylko 4 kolory (mówiąc ściśle 3, bo czwarty kolor jest wspólny dla całego ekranu). Do zapisu ekranu używa się 8 KB mapy bitowej + 1 KB pamięci ekranu (która w tym momencie jest mapą kolorów) + 1 KB młodszych *nibble* o adresach od $d800-$dc00. Za to, z której mapy kolorów ma być pobierany kolor danego punktu, odpowiadają dwa bity. Ustawienie bitów na 00 równa się użyciu koloru tła ($d021), ustawienie na 11 — koloru określanego przez *nibble* z $d800, na 01 — starszego *nibble* z pamięci kolorów ($0400), i na 10 — młodszego *nibble* z pamięci kolorów. Interpretacja pamięci ekranu następuje w sposób przedstawiony na rysunku 1.

Czy można spowodować, aby komputer mógł wyświetlać więcej kolorów w bliskim sąsiedztwie? Spróbujmy się nad tym zastanowić. Gdyby tak co linię rastra zmieniać zawartość pamięci kolorów... Niestety jest to nierealne, gdyż trzeba by przepisywać za każdą linią 40 bajtów, a przy częstotliwości 1 MHz byłoby to po prostu niemożliwe.

No tak, ale przecież wcale nie trzeba nic przepisywać. Wystarczy po każdej linii zmieniać komórkę wskazującą lokalizację mapy kolorów ($d018). W takim wypadku na zapamiętanie ekranu potrzeba: 8 KB (mapa bitowa) + 8 KB (8∗1 KB mapy kolorów) + 1 KB *nibble* $d800-$dc00. Mapę bitową i mapy kolorów musimy umieścić w tym samym banku VIC. Do wyboru mamy banki $4000-$7fff i $c000-$ffff, pozostałe banki nie nadają się ze względu na to, że nie możemy wykorzystać pamięci leżącej „pod" generatorem znaków. W ten sposób mamy jeden kolor wspólny dla całego ekranu (bity ustawione na 00), jeden kolor wspólny dla każdego „znaku" (bity ustawione na 11) i po dwa niezależne kolory dla każdego bajtu (bity ustawione na 10 i 01). Sposób interpretacji pamięci na ekranie przedstawia rysunek 2.

Ten trik nazywa się FLI (Flexible Line Interpretation), co oznacza „zmienna interpretacja linii" (został on wymyślony przez członków holenderskiej grupy BLACKMAIL w roku 1989 i pierwszy raz użyty w demie pt. SOPHISTICATED 3). Aby wyświetlać grafikę w trybie FLI, potrzebna jest specjalna procedura, której przykładowy listing zamieszczamy obok.

```
         *= $1000
;---------------------------------------
;     fli-shower by jetboy/parados
;      (c) commodore & amiga 1992
;---------------------------------------
         ldy #$00       ;tworzenie tabelek
tabgen   tya            ;dla d011 i d018
         clc
         adc #$01
         and #$07
         ora #$38
         sta tab11,y
         tya
         adc #$01
         asl a
         asl a
         asl a
         asl a
         ora #$08
         and #$7f
         sta tab18,y
         iny
         cpy #$c8
         bne tabgen
;---------------------------------------
         sei             ;procedura
         lda #$70        ;inicjalizacji
         sta tab11+199   ;przerwan
         lda #$08        ;oraz ustawenie
         sta tab18+199   ;wartosci
         ldx #<irq1      ;poczatkowych
         ldy #>irq1
         stx $0314
         sty $0315
         lda #$18        ;multicolor
         sta $d016
         lda #$00        ;kolory ramki
         sta $d020       ;i papieru
         sta $d021
         lda #$3b        ;tryb graficzny
         sta $d011
         lda #$7f
         sta $dc0d
         lda $dd00       ;drugi bank VIC
         and #$fc        ;($4000-$8000)
         ora #$02
         sta $dd00
         lda #$01
         sta $d01a
         lda #$7f
         sta $dc0d
         lda #$2e
         sta $d012
         lda #$1b
         sta $d011
         ldx #$00        ;przepisanie
loop     lda $3c00,x     ;danych do pamie-
         sta $d800,x     ;ci kolorow
         lda $3d00,x
         sta $d900,x
         lda $3e00,x
         sta $da00,x
         lda $3f00,x
         sta $db00,x
         inx
         bne loop
         cli
         jmp *
;---------------------------------------
irq1     lda #<irq2     ;pierwszy"poziom"
         sta $0314      ;przerwan
         lda #>irq2
         sta $0315
         ldy #$0a
         dey
         bne *-1
         cli
;---------------------------------------
irq2     ldx #$00       ;drugi "poziom"
mainloop lda tab18,x    ;przerwan +
         sta $d018      ;glowna petla
         lda tab11,x    ;wyswietlajaca
         sta $d011
         inx
         cpx #200
         bne mainloop
;---------------------------------------
         inc $d019
         lda #<irq1
         sta $0314
         lda #>irq1
         sta $0315
         pla
         pla
         pla
         pla
         pla
         pla
         jmp $ea81
;---------------------------------------
         *= $1100
tab18    .byte 0
         *= $1200
tab11    .byte 0
;---------------------------------------
```

Niestety nie ma róży bez kolców, zyskując na kolorach tracimy mnóstwo czasu na wyświetlanie obrazka. Poza tym, tryb FLI nie daje się uzyskać w 3 pierwszych kolumnach, w wyniku czego mamy do dyspozycji rozdzielczość 148x200 punktów. Po prostu, aby móc skutecznie zmieniać rejestr $d018, musimy w każdej linii najpierw „popchnąć" komórkę $d011. Gdyby tego nie robić, to wprawdzie zawartość tego rejestru zmieniałaby się, lecz efekty można by było zaobserwować dopiero co ósmą linię. Manipulacja komórką $d011 ma jeszcze jedną zaletę, mianowicie

likwiduje ona problem z cyklowaniem — wszystkie linie są teraz „krótkie” i dzięki temu można napisać zwykłą pętlę bez żadnych specjalnych kombinacji (jeżeli ktoś nie rozumie, co to jest krótka linia, to odsyłam do znakomitych artykułów Bartka Kachniarza z cyklu „Pamiętnika artylerzysty”). Przykładowy rysunek w trybie FLI narysowany przez OGAMI-ego z grupy Fairlight ilustruje zdjęcie.

Do tej pory powstało kilka programów służących do rysowania w trybie FLI, do najlepszych należą:

**Fli designer/CPU**
Nie ma on zbyt dużych możliwości, jest jednak zdecydowanie najwygodniejszy ze względu na dzielony na pół ekran. W górnej części widzimy fragment przygotowywanego obrazka, a w dolnej powiększony fragment ekranu roboczego.

**Fligraph v2.2/BLACKMAIL**
Edytor ten wyróżnia się tym, iż możliwe jest wykorzystanie trzech pierwszych kolumn. Wprawdzie nie można na nich używać trybu FLI, lecz tylko *multicolor*. Programu tego używa Hein Holt, jeden z najwybitniejszych grafików tworzących na C-64 (jest on m.in. twórcą znakomitych obrazków z części dema Dutch Breeze/BLACKMAIL pt. THE COLECTION).

**Quickdraw/FACES**
Posiada szczególnie dużo opcji np. możliwość tworzenia okręgów, linii, prostokątów itd.

**Fli editor v3.0/DOLPHINS**
Z pozoru jest to najbardziej prymitywny edytor do tworzenia grafiki w trybie FLI. Ma on jednak jedną wielką zaletę: można w pełni kontrolować koloru, miejsce, w którym go użyjemy i przez jaką kombinację bitów będzie on reprezentowany w pamięci. Jest to czasem bardzo istotne, szczególnie gdy chcemy robić takie efekty jak FLI Tech-Tech, lub PLOTER nad FLI (można je zobaczyć np. w demach polskiej grupy PARADOS pt. ONE NOP TOO FAR i PIECE OF SHIT).

*RAFAŁ PIASEK*

**SŁOWNICZEK:**

- **NIBBLE** — inaczej połbajt, np. jeżeli mamy w bajcie zapisaną wartość $1F, to 1 będzie starszym *nibble*, a F młodszym.
- **KRÓTKA LINIA** — zobacz serię artykułów Bartłomieja Kachniarza pt. „Pamiętnika artylerzysty”.
- **TECH-TECH** — efekt polegający na tym, że każda pozioma linia „buja się” niezależnie od innych w prawo i w lewo.
- **PLOTER** — efekt polegający na tym, że po ekranie porusza się gromada punktów po torze mającym mniej lub więcej wspolnego z funkcją sinus (ewentualnie cosinus).



*Takiej grafiki nie „powstydziłaby się” Amiga — oto co można zrobić na C-64 za pomocą trybu FLI. (Jeśli chcesz ujrzeć taki obrazek u siebie w domu, zamów dyskietkę C&A nr 10!)*

| | | | |
|---|---|---|---|
| 00 | 01 | 10 | 11 |
| 00 | 01 | 10 | 11 |

...(itd.)

```
00 - $d020
01 - młodszy nibble $0400+w*40+k
10 - starszy nibble $0400+w*40+k
11 - $d800+w*40+k
```

*Sposób interpretacji fragmentu mapy bitowej odpowiadającej położeniem znakowi o współrzędnych k,w jako obraz w trybie multicolor.*

| | | | |
|---|---|---|---|
| 00 | 01 | 10 | 11 |
| 00 | 01 | 10 | 11 |
| 00 | 01 | 10 | 11 |
| 00 | 01 | 10 | 11 |
| 00 | 01 | 10 | 11 |
| 00 | 01 | 10 | 11 |
| 00 | 01 | 10 | 11 |
| 00 | 01 | 10 | 11 |

```
01 - młodszy nibble $4000+w*40+k
10 - starszy nibble $4000+w*40+k
01 - młodszy nibble $4400+w*40+k
10 - starszy nibble $4400+w*40+k
01 - młodszy nibble $4800+w*40+k
10 - starszy nibble $4800+w*40+k
01 - młodszy nibble $4c00+w*40+k
10 - starszy nibble $4c00+w*40+k
01 - młodszy nibble $5000+w*40+k
10 - starszy nibble $5000+w*40+k
01 - młodszy nibble $5400+w*40+k
10 - starszy nibble $5400+w*40+k
01 - młodszy nibble $5800+w*40+k
10 - starszy nibble $5800+w*40+k
01 - młodszy nibble $5d00+w*40+k
10 - starszy nibble $5d00+w*40+k

(00 - $d020, 11 - $d800+w*40+k)
```

*Sposób interpretacji fragmentu mapy bitowej, odpowiadającej położeniem znakowi o współrzędnych k,w jako obraz w trybie FLI.*

# ASEMBLER 6502 (cz.V)

## ROLOWANIE BAJTÓW I NIE TYLKO

Zawartość komórek możemy nie tylko dodawać lub odejmować (co już od dobrego miesiąca umiemy). 6502 daje nam także bogate możliwości PRZESUWANIA bitów w obrębie jednego bajtu. Wiemy, że bajt składa się z ośmiu bitów, z których każdy może mieć wartość 0 lub 1. Jeżeli ułożymy je wszystkie w jednej linii, możemy je potraktować jako osiem dwójkowych rzędów wielkości. Weźmy jakąkolwiek liczbę. Niech to będzie dla przykładu 105. Po przełożeniu tego na układ dwójkowy dostaniemy liczbę 01101001. Jeśli przesuniemy to co mamy o jeden bit w lewo (po prawej stronie wstawiając 0), dostaniemy liczbę 11010010. Krótkie obliczenia (128+64+16+2)... wynik: 210! Czyli dokładnie dwa razy więcej! Oznacza to, że przesunięcie bitów w lewo jest równoznaczne z pomnożeniem danej liczby przez dwa. Czy można wyciągnąć z tego wniosek, że przesunięcie bitów w prawo to nic innego jak podzielenie liczby przez dwa? Właściwie tak. Każdy bit przekaże swoją zawartość sąsiadowi po prawej. A wiemy, że sąsiad po prawej ma dwa razy mniejszą wartość liczbową. Dlatego też możemy podejrzewać, że zaiste tak będzie.

Weźmy więc znowu 105 (czyli %01101001, procentem programiści zwykli poprzedzać liczby zapisane dwójkowo, podobnie jak dolarem — szesnastkowo) i przesuńmy w prawo (po lewej wstawiając 0). Mamy więc %00110100, czyli (32+16+4) 52. A przecież 105 na 2 to 52,5! Co stało się z brakującymi 0,5? Przyjrzyjmy się naszym liczbom dwójkowym. W pierwszej mamy cztery jedynki, w drugiej zaś już tylko 3. Wniosek z tego taki, że nie możemy ot tak, po prostu wyrzucać na śmietnik niepamięci bitów, które wypadły nam poza bajt podczas przesuwania. Trzeba je gdzieś wstawiać. Potrzebny jest nam choć jeden bit, do którego łatwo będzie się dostać.

Jak znalazł jest tutaj flaga przeniesienia (bit C, czyli Carry). Przeniesienie przecież po to właśnie istnieje, by wstawiać doń bity, które „wyskoczyły" nam podczas jakichś operacji arytmetycznych poza przewidziane osiem bitów. Tak właśnie działają rozkazy ASL i LSR.

ASL to przesunięcie zawartości bajtu w lewo (Arithmetic Shift Left). Jeżeli wykonamy je na jakiejś komórce, bity 0 — 6 przesuną się o jedno oczko w lewo. Jako nowy bit 0 wstawione będzie 0. Jeśli stary bit 7 był jedynką, flaga C zostanie zapalona, jeśli zaś był zerem — zgaszona. Mam nadzieję, że klarownie objaśnia to rys. 1.

Sprawdźmy, jak rozkaz ASL działa w praktyce. Musimy napisać program, który wstawi do pamięci jakąś wartość, następnie przesunie ją w lewo i da nam jakoś znać, jeśli znacznik C został zapalony.

| MONITOR | ASEMBLER |
|---|---|
| | *=$2710 |
| A 2710 LDA #$40 | LDA #$40 |
| A 2712 STA $2800 | STA $2800 |
| A 2715 ASL $2800 | ASL $2800 |
| A 2718 LDA #$00 | LDA #$00 |
| A 271a BCC $271e | BCC ZERO |
| A 271c LDA #$01 | LDA #$01 |
| A 271e STA $2801 | ZERO STA $2801 |
| A 2721 BRK | BRK |

Uruchom program przez SYS 10000 albo G2710. W komórce $2800 (dziesiętnie 10240) znajdzie się wynik przesunięcia, w $2801 (10241) zaś to, co znalazło się w znaczniku przeniesienia. Jeśli wpiszemy program tak, jak jest podany — w $2800 dostaniemy $80, a w $2801 — $00, czyli tak, jak to było do przewidzenia. Wpiszmy jednak w pierwszej linii LDA #$ff, a potem uruchomimy program. $ff to %11111111. Po przesunięciu w lewo dostaniemy wynik $fe, czyli %11111110. W komórce $2801 (czyli 10241) będzie jedynka, która bierze się z ostatniej jedynki bajtu poprzedniego. Powinna być więc traktowana jako dodatkowy bit przesuwanego przez nas bajtu. Proponuję trochę się pobawić tym programikiem — pomoże to dokładnie pojąć działanie rozkazu ASL.

Instrukcją odwrotną jest LSR. LSR to przesunięcie w prawo (Logical Shift Right). Ten rozkaz przesuwa bity w bajcie w prawo, przy czym skrajny prawy bit jest wstawiany do C, zaś za skrajny lewy komputer bierze 0. Działanie tego rozkazu objaśnia rys. 2 oraz ten oto niewielki programik:

| MONITOR | ASEMBLER |
|---|---|
| | *=$2710 |
| A 2710 LDA #$40 | LDA #$40 |
| A 2712 STA $2800 | STA $2800 |
| A 2715 LSR $2800 | LSR $2800 |
| A 2718 LDA #$00 | LDA #$00 |
| A 271a BCC $271E | BCC ZERO |
| A 271c LDA #$01 | LDA #$01 |
| A 271e STA $2801 | ZERO STA $2801 |
| A 2721 BRK | BRK |

Po uruchomieniu programu — zgodnie z przewidywaniami — w komórce $2800 znajdzie się $20 (32), czyli dwa razy mniej niż $40 (64). Wszystko się więc zgadza.

Para opisywanych rozkazów okazała się jednak niewystarczająca. Nie dla wszystkich bowiem zastosowań za nowy, skrajny bit można było przyjąć 0. Nie byłoby też szczególnie wygodne, gdyby zaprojektować dwa rozkazy, które wstawiałyby 1 zamiast 0. Dla większej elastyczności i wygody powstała więc dwójka ROL i ROR.

ROL to ROtate Left, czyli obrócenie w lewo. Od rozkazu ASL różni się tym, że jako skrajny prawy bit wstawiane jest nie — 0, lecz to, co było w C PRZED wykonaniem rozkazu. Zaś to, co było na lewym skraju, wchodzi do C (tak jak w ASL). Ilustruje to rysunek 3 oraz ten oto program:

| MONITOR | ASEMBLER |
|---|---|
| | *=$2710 |
| A 2710 SEC | SEC |
| A 2711 LDA #$40 | LDA #$40 |
| A 2713 STA $2800 | STA $2800 |
| A 2716 ROL $2800 | ROL $2800 |
| A 2719 LDA #$00 | LDA #$00 |
| A 271b BCC $271e | BCC ZERO |
| A 271d LDA #$01 | LDA #$01 |
| A 271f STA $2801 | ZERO STA $2801 |
| A 2722 BRK | BRK |

W komórce $2800 znajdzie się $81, czyli %10000001. $40 to %01000000. Jedynka, która została przeniesiona z szóstego do siódmego bitu a to, co było w przeniesieniu, pojawiło się po prawej. Jeżeli napiszemy CLC zamiast SEC, wynik będziemy mieli identyczny, jak w przypadku ASL — jako bit zerowy będzie 0.

Rozkazem, który działa w stronę przeciwną jest ROR. ROR to skrót od angielskich słów ROtate Right, czyli obróć w prawo. Jak wskazuje nazwa, przesuwa on zawartość bajtu w prawo. Poza tym w miejsce bitu siódmego wstawia zawartość Carry, a to, co było w bicie zerowym, do przeniesienia odsyła. I bardzo dobrze (rys. 4).

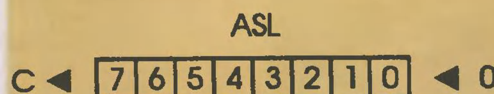
| MONITOR | ASEMBLER |
|---|---|
| | *=$2710 |
| A 2710 SEC | SEC |
| A 2711 LDA #$40 | LDA #$40 |
| A 2713 STA $2800 | STA $2800 |
| A 2716 ROR $2800 | ROR $2800 |
| A 2719 LDA #$00 | LDA #$00 |
| A 271b BCC $271e | BCC ZERO |
| A 271d LDA #$01 | LDA #$01 |
| A 271f STA $2801 | ZERO STA $2801 |
| A 2722 BRK | BRK |

Po wykonaniu w $2800 dostaniemy z $40 (czyli %00010000) liczbę $a0 (%10001000). W $2801, zgodnie z przewidywaniami będzie 0 — bit zerowy był zgaszony.

Oczywiście, rozkazy te służą nam nie tylko do przesuwania jakichś abstrakcyjnych bajtów w prawo czy w lewo. Mogą się nam przydać, jeśli np. będziemy chcieli sprawdzić po kolei bity danego bajtu, jeżeli będziemy chcieli zrobić procedurę płynnego przesuwania napisów na ekranie graficznym albo mnożyć/dzielić coś przez wielokrotności 2.

*BARTŁOMIEJ KACHNIARZ*

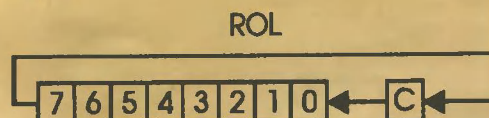
Rys. 1



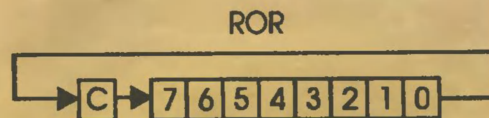

Rys. 2




Rys. 3




Rys. 4



**Komputery C-16/116 128/PLUS/4 zostały wyposażone w bardzo dobre interpretery języka BASIC, zawierające między innymi zestaw poleceń ułatwiających samodzielne tworzenie grafiki. Co ważniejsze, nie musisz być zaawansowanym programistą, aby móc porysować sobie po ekranie — wprost przeciwnie, taka zabawa może być początkiem głębszej znajomości z komputerem. Specjalnie dla początkujących („zielonych" — jak część z Was określa się w listach) zaczynam od podstaw.**

# GRAFIKA DLA KAŻDEGO

Na początek trochę danych technicznych: nasze komputery wyświetlają obraz z rozdzielczością 200 poziomych linii po 320 punktów (a więc na całym ekranie mamy 64 000 punktów i każdy z nich może być niezależnie od innych zapalony bądź zgaszony). Komputery z rodziny C-16 potrafią wygenerować 121 kolorów. C-128 — tylko 16, ale umożliwia zdefiniowanie ośmiu niezależnych obiektów ruchomych (ang. sprite) i posiada dodatkowy tryb tekstowy pozwalający na wyświetlanie 80-ciu znaków w linii (przy okazji: w tym trybie nie można korzystać z grafiki wysokiej rozdzielczości).

Teraz możemy rozpocząć część praktyczną naszego spotkania. Po włączeniu komputera napisz GRAPHIC 2,1 i naciśnij klawisz RETURN (ten klawisz musisz naciskać zawsze, gdy chcesz, by komputer wykonał polecenie znajdujące się w linii z kursorem). Po wykonaniu polecenia napis READY i kursor znajdą się u dołu ekranu. Na razie tego nie widać, ale właśnie podzieliłeś ekran na dwa obszary: górna część jest przeznaczona do wyświetlania obrazu wysokiej rozdzielczości (czyli naszych rysunków), a ostatnie pięć linijek w dalszym ciągu służy do wyświetlania normalnych znaków (dzięki temu będzie widać wpisywane przez nas polecenia.

Narysujmy teraz na środku ekranu kropkę: wpisz DRAW 1,160,100. Jest! Jedynka w wydanym przed chwilą poleceniu oznaczała, że kropka ma być narysowana. Gdyby było zero, to zostałaby wytarta (ta zasada obowiązuje także przy innych poleceniach graficznych). Spróbuj wytrzeć narysowany przed chwilą punkt. Zastanawiasz się, co oznaczają pozostałe liczby? Informują one komputer, którego miejsca na ekranie dotyczy polecenie. Przyjęto, że każdy punkt ekranu można opisać z pomocą układu współrzędnych, czyli dwóch osi, na których odpowiednio odczytujemy liczby (tak, jak na rysunku).

Nasz punkt (160,100) znajduje się pośrodku ekranu, ponieważ (jak już wspominałem) ekran ma 320 na 200 punktów.

Narysujmy teraz okrąg, którego środek będzie się znajdował w punkcie o współrzędnych 160,50 a długość promienia będzie wynosiła 40:

**CIRCLE 1,160,50,40**

Tym samym poleceniem można narysować elipsę (a także dowolne wielokąty — ale o tym innym razem) — trzeba tylko podać drugi promień:

**CIRCLE 1,160,50,40,20**

Sprawdź, co się stanie, gdy zamienisz miejscami dwie ostatnie liczby. Narysuj kilka okręgów i elips w różnych miejscach. Możesz modyfikować polecenie, które już wpisałeś i które jeszcze nie schowało się pod ekran graficzny — wystarczy ustawić kursor na liczbie, którą chcesz zmienić, wpisać nową wartość i wcisnąć RETURN. Prawda, że proste?

Możesz także rysować linie: należy podać współrzędne punktów będących początkiem i końcem rysowanej kreski. Polecenie rysujące kreskę od lewego-górnego rogu do środka ekranu ma postać

**DRAW 1,0,0 TO 160,100**

a ze środka do drugiego górnego rogu:

**DRAW 1,160,100 TO 319,0**

Jednym poleceniem można narysować trójkąt:

**DRAW 1,50,50 TO 60,60 TO 60,40 TO 50,50**

W podobny sposób można by narysować kwadrat lub prostokąt, lecz do tego celu służy instrukcja BOX. Podajemy lewy-górny, a następnie prawy-dolny róg naszego prostokąta i to wystarczy, by komputer zrozumiał, o co nam chodzi, np.:

**BOX 1,30,50,80,75**

Dodając na końcu jeszcze jedną liczbę otrzymamy ten sam prostokąt, ale obrócony o 45 stopni:

**BOX 1,30,50,80,75,45**

Wypróbuj te komendy. Jeżeli chcesz wytrzeć ekran, użyj polecenia SCNCLR.

Teraz wykorzystamy fakt, że komputer można bardzo łatwo zmusić do powtarzania pewnych czynności i narysujemy dziesięć współśrodkowych okręgów o wzrastających promieniach. Najwygodniej będzie wykorzystać w tym celu pętlę FOR... NEXT, np. **FOR R=5 TO 55 STEP 5:CIRCLE 1,160,100,R:NEXT R**

Ciekawszy efekt otrzymamy nieco komplikując poprzedni przykład. Wyczyść ekran (komendą SCNCLR) i napisz:

**FOR R=0 TO 50 STEP 5:CIRCLE 1,160,100,R,50-R:NEXT R**

Do tej pory dolna część obrazu była „ucięta" przez linie z tekstem. Niedogodności tej pozbędziemy się włączając pierwszy tryb graficzny. Tym razem nie będzie widać wpisywanych przez nas poleceń, dlatego ponumerujemy je i wpiszemy do pamięci. Powstały w ten sposób program sam na początku włączy ekran graficzny, a po zakończeniu rysowania poczeka na wciśnięcie jakiegokolwiek klawisza i powróci do trybu tekstowego (tryb graficzny 0). Wpisz kolejne linie programu:

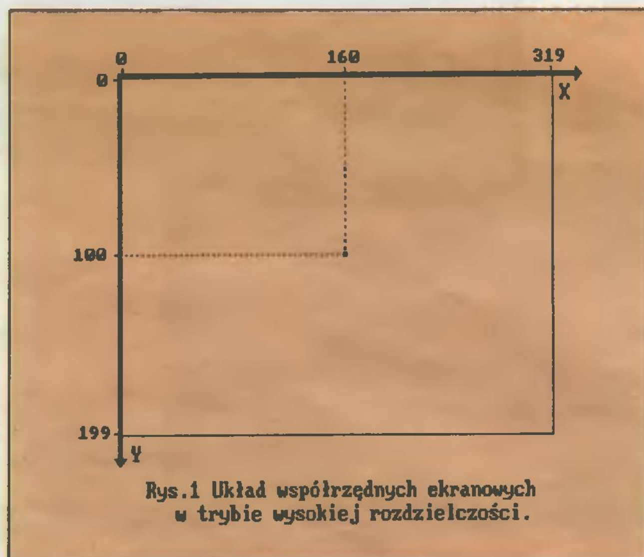
```
10 GRAPHIC 1,1
20 FOR A=0 TO 180 STEP 3
30 BOX 1,160-A,100-A,160+A,100+A,A
40 NEXT A
50 GETKEY A$
60 GRAPHIC 0
```

Teraz wpisz RUN i program zostanie uruchomiony. W porównaniu ze stosowaną przez nas do tej pory metodą wpisywanie programu do pamięci daje nam do ręki kilka atutów: mamy więcej miejsca na polecenia dla komputera, bez problemu korzystamy z pełnego ekranu graficznego i możemy wprowadzać poprawki do programu bez obawy, że jego tekst ucieknie nam z ekranu. Treść programu w każdej chwili można przywołać z pamięci na ekran poleceniem LIST.

Zmień liczby w linii 20 i zaobserwuj, co się zmieni w rysunku (np. zmień liczbę 3 na 1). Na razie proponuję Ci pobawić się poznanymi dzisiaj poleceniami, a za miesiąc przedstawię listę wszystkich instrukcji graficznych i przejdziemy do bardziej skomplikowanych programów.

(cdn.)

*WOJCIECH KAZIMIERCZAK*



Rys.1 Układ współrzędnych ekranowych w trybie wysokiej rozdzielczości.

*Ta grafika została zrobiona za pomocą tylko jednej (widocznej na zdjęciu) linii programu...*



# BLACK CRYPT

Dwadzieścia lat temu, w krainie zwanej Astera, jeden z członków bractwa duchownego, Estoroth Paingiver, zaczął używać czarów do czynów niegodnych miana Wielkiego Duchownego. Rada Czterech skazała go zatem na banicję. Urażony tym do żywego Paingiver obiecał wrócić i zemścić się. I tak się stało. Armia najstraszniejszych demonów pustoszyła ziemię Astery pod jego dowództwem. Nikt nie mógł go pokonać. Dopiero połączone siły czterech bractw (Wojowników, Duchownych, Czarodziejów i Druidów) pozwoliły wytworzyć odpowiednie bronie i czary mogące pokonać złe moce.

Zadanie to powierzono czterem mistrzom, najsilniejszym i najszlachetniejszym, wybranym po jednym z każdego bractwa. Olbrzymim kosztem krwi i potu udało się pokonać armię demonów, a samego Paingivera wpędzono do lochów zburzonego zamku, gdzie dzięki nowym broniom zamrożono go i wepchnięto w otchłań czasu. Nie doceniano jednak mocy Estorotha. Po latach okazało się, że otchłań czasu zamyka się, a zamrażacz stracił swoją moc. Rada Czterech postanowiła wysłać kolejną wyprawę, aby jeszcze raz poskromić Estorotha Paingivera.

I tu właśnie zaczyna się Twoje zadanie. Masz do dyspozycji czterech mistrzów. Musisz spenetrować lochy starego zamku w poszukiwaniu magicznych broni pozostawionych tu przez poprzednią wyprawę. Pozwolą Ci one wykonać zadanie, a następnie odnaleźć Czarną Kryptę, w której ukrywa się Paingiver. Podczas poszukiwań możesz się natknąć na różne wielce przydatne przedmioty (broń, żywność, części opancerzenia, amulety, księgi z zaklęciami) i bardzo nieprzyjemne w postaci niedobitków armii demonów. Sam labirynt jest bardzo ciekawy, ściany do przejścia w jedną stronę, magiczne pola i ryty zagradzające drogę, pułapki, dziury w podłodze oraz teleporty.

W sumie jest do przejścia 28 poziomów i nieźle trzeba się nagłowić, aby dotrzeć do końca, bowiem poziomy są ze sobą związane poprzez przełączniki i płyty naciskowe. Czasem trzeba coś przełączyć na jednym poziomie, aby móc poruszać się bez przeszkód w drugim.

Grafika gry jest opracowana perfekcyjnie. Z lochu bije chłodem, a potwory w nim spotykane mogą spokojnie startować w horrorowym konkursie piękności. Napotkane przedmioty swoim wyglądem wyjaśniają swoje przeznaczenie, więc nie ma mowy o pomyłkach. Również świetnie obmyślone jest sterowanie. Bardzo mi się podoba, że wskaźnik myszy zmienia się w rzecz, na której klikniemy. Ułatwia to przenoszenie i sprawdzanie przedmiotów.

Bardzo przyjemną niespodzianką jest instrukcja, oczywiście całkowicie w języku polskim. Zawiera ona wszystko co niezbędne do rozpoczęcia i zakończenia gry. Szczegółowo wyjaśnia sposób rozpoczęcia gry, który polega na stworzeniu grupy czterech śmiałków, łącznie z wybraniem ich twarzy i określeniem ich charakterystyk. Objaśnia co znaczą poszczególne wskaźniki oraz jak poruszać się, walczyć z potworami, używać czarów, odżywiać się i odpoczywać. Sam opis czarów zajmuje dwie kartki. A jak ważne są tajemne moce w takich grach nie trzeba nikomu wyjaśniać.

W instrukcji znajduje się też cała historia Czarnej Krypty (spisana ręką Maatoka) oraz charakterystyka czterech bractw, z której należy wyciągnąć wnioski co do uzbrojenia i opancerzenia poszczególnych postaci. Bowiem nie każdy z mistrzów potrafi efektywnie posługiwać się każdą bronią i nosić każdy rodzaj pancerza. Instrukcja zawiera również „ostatnią deskę ratunku", czyli plany wszystkich poziomów z zaznaczeniem przedmiotów jakie można znaleźć i lokalizacje teleportów, przełączników i płyt naciskowych — jest to o tyle ważne, że niektóre z nich są niewidoczne.

A teraz trochę objaśnień: na ekranie, na którym tworzymy swoją grupę, zastosowano skróty nazw angielskich poszczególnych atrybutów postaci. Są to: STRengh — siła, INTelligence — inteligencja, WISdom — mądrość, DEXterity — zręczność i CONstitution — to zostało przetłumaczone jako „konstytucja fizyczna", ale lepszym określeniem byłoby chyba „budowa fizyczna" lub po prostu „muskulatura". Właściwe określenie tych atrybutów jest bardzo ważne. Postacie będące w pierwszym szeregu (wojownik i duchowny) powinny mieć dużą siłę i zręczność, gdy skutecznie walczyć z potworami, natomiast postacie używające czarów — mądrość i inteligencję, aby szybciej uczyć się sztuki magicznej.

Drugi szereg należy uzbroić w łuki i sztylety, aby w razie starcia walczyć w miarę możliwości wszystkimi postaciami. Więcej punktów zdobytych w walce to szybsze osiągnięcie następnego poziomu doświadczenia, a przez to możliwość korzystania z silniejszych czarów i większa sprawność w walce. Czar Wizard Sight i Compass nie działają wiecznie. Należy je odnawiać co jakiś czas, aby mieć narysowaną automatycznie całą mapę poziomu. Jeżeli wykorzystamy (rzucimy) jakiś czar, to należy nauczyć się go jeszcze raz, aby mieć go „pod ręką" w razie potrzeby. Niektóre czary, zwłaszcza znalezione na zwojach, wymagają od rzucającego odpowiedniego poziomu doświadczenia. Utracone w walce punkty odporności można odzyskać czarami lub odpoczynkiem. Podczas odpoczynku należy zostawić jednego z członków grupy na warcie, korzystać z fontann jako źródła wody, napoić grupę do oporu i uzupełnić zapasy w bukłakach. Każdy znaleziony zwój z informacjami należy potraktować czarem wyjawiającym prawdę, bowiem złośliwy Paingiver podłożył mnóstwo fałszywych informacji.

Nie należy skakać poprzez dziury w podłodze. Kończy się to w większości przypadków utratą przynajmniej połowy grupy. W przypadku utraty członka ekipy nie należy rozpaczać. Posiadając Klejnot Śmierci można go wskrzesić odpowiednim zaklęciem. Czerwonego potwora z dwoma głowami można unieszkodliwić tylko czymś co się nazywa „orgeblade".

Ponadto trzeba przełączać i otwierać wszystko co się da, włazić w każdy zakamarek lochów, bo nie wiadomo czy na końcu ślepego korytarza nie ma niewidzialnej płyty naciskowej otwierającej drogę na innym poziomie. Półprzezroczyste bariery mieniące się różnokolorowo to Magiczne Pola — trzeba je odczarować, a podobne, tylko fioletowe, to teleporty. Natomiast czerwone plamy w korytarzu to ryty, które również trzeba odczarować, aby bezpiecznie przejść.

Najważniejsze cudowne bronie to: Vortex (wojownik), Force Hammer (duchowny), Soul Freezer (czarodziej) i Protector (druid). Ich trzeba szukać w pierwszej kolejności. Bez nich nie mamy szans z Paingiverem.

Podsumowując, gra BLACK CRYPT jest jedną z najlepszych w klasie D&D (Dungeons & Dragons). Wymaga poruszania szarych komorek, a jednocześnie nie jest pozbawiona części zręcznościowej. Wspaniała grafika 3D i przejrzyste sterowanie stanowią jej dodatkową zaletę. Nie mówiąc już o polskiej instrukcji, bez której nie da się pograć z choćby z powodu różnorodności czarów lub konieczności udzielania prawidłowych odpowiedzi na pytania w czasie gry.

*PAWEŁ GALAS*

**FIRMA: Electronics Arts**
**DYSTRYBUTOR: IPS Computer Group, Warszawa**
**RODZAJ GRY: przygodowa D&D**
**KOMPUTER: Amiga**
**WYMAGANIA: -**

| ocena | 0 100% |
|---|---|
| grafika | ■■■■■■ |
| muzyka | □□□ |
| ogólnie | ■■■■■■■ |



Black Crypt



Piracy


Escape from the planet of the robot monsters

# GRY

## SLAYER

Weź w rękę joystick i mocno ściśnij go w dłoni. Masz raczej twardy orzech do zgryzienia. Musisz brawurowo przelecieć przez wrogie umocnienia, pokonać wielu napastników unikając jednocześnie groźnych skał i innych przeszkód. Prawie wszystko, czego dotknie Twój statek, przynosi Ci natychmiastową zgubę. Szczególnie zdradliwe są działka laserowe — potrafią one zupełnie niespodziewanie zmienić kierunek ostrzału, co zwykle kończy się dla Ciebie tragicznie.

Gra ma dość dobrą grafikę, tyle że ciężko jest zwrócić na to uwagę w lawinie ognia, jaką serwuje nam przeciwnik. Podobnie jest zresztą z dźwiękiem — wszelkie rozproszenie uwagi kończy się stratą pojazdu.

SLAYER jest typową grą typu ZABIJ-WSZYSTKO, różniącą się od pozostałych jedynie liczbą przeciwników do unicestwienia.

*BIONEK*

**FIRMA: Hewson Consultants Limited**
**DYSTRYBUTOR: JTT Computer S.C. Wrocław**
**RODZAJ GRY: zręcznościowa**
**KOMPUTER: Amigs**
**WYMAGANIA: -**

| ocena | 0 100% |
|---|---|
| grafika | ▮▮ |
| muzyka | ▯▯ |
| ogólnie | ▮ |

## BUFFALO BILL'S RODEO GAMES

Czy myślałeś już o tym, aby pojechać na Dziki Zachod? Jeśli tak, to przed wyjazdem powinieneś zagrać w BUFFALO BILL'S RODEO GAMES, czyli wziąć udział w olimpiadzie dla prawdziwych mężczyzn.

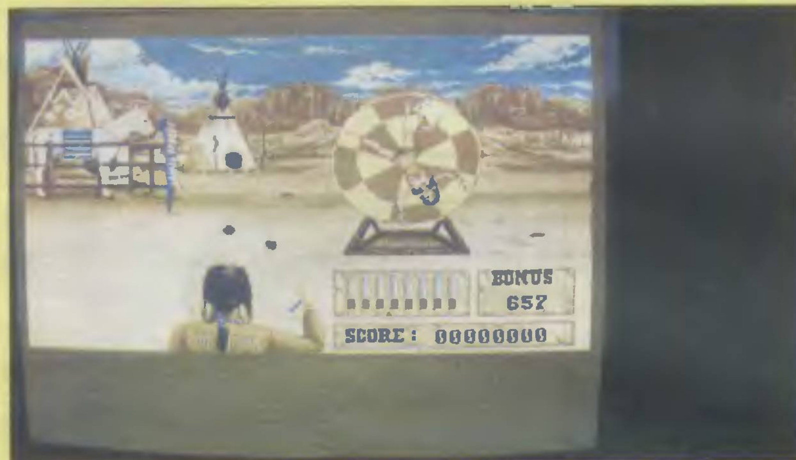
Aby ukończyć grę, będziesz musiał rzucać nożem do tarczy, do której przywiązana jest kobieta, strzelać do pojawiających się napastników, oszczędzając jednocześnie spokojnych mieszkańców, szeryfów i ludzi odwróconych do Ciebie plecami. Będziesz także musiał ujeździć rozszalałego konia i złapać byka na lasso. Są także jeszcze inne dyscypliny...

Program ten można zaliczyć do szerokiej gamy gier zręcznościowych. Sam pomysł nie jest nowy, lecz ma w sobie coś przyciągającego. Wykonanie oceniam jako przeciętne, choć np. grafice w zasadzie nic nie można zarzucić.

*BIONEK*

**FIRMA: Tynesoft Computer Software**
**DYSTRYBUTOR: IPS Computer Group, Warszawa**
**RODZAJ GRY: zręcznościows**
**KOMPUTER: Amiga 500**
**WYMAGANIA: -**

| ocena | 0 100% |
|---|---|
| grafika | ▮▮▮▮▮▮▮ |
| muzyka | ▯▯▯ |
| ogólnie | ▮▮ |




*Buffalo Bill's Rodeo Games*



*Slayer*

## ESCAPE FROM THE PLANET OF THE ROBOT MONSTERS

Znajdujesz się na Planecie X w kompleksie wielu fabryk opanowanych przez roboty zombi. Zadanie jakie przed Tobą stoi jest dość trudne, gdyż musisz ocalić profesora Sarah Belluma i uratować zniewolonych zakładników zanim staną się cyborgami. Przechodząc przez fabryki, niszcząc roboty, uwalniając niewolników musisz pamiętać, że Twoja energia jest ograniczona. Pod ścianami stoją pojemniki zawierające żywność, bomby i paliwo uzupełniające energię. Niektóre ze zniszczonych robotów pozostawiają zielone kryształy, które także potęgują Twoją siłę. Oszczędzaj bomby, gdyż podążając coraz dalej napotykać będziesz złych Reptilonów, których zniszczyć można będzie jedynie ciągłym ogniem i bombardowaniem.

Aby uratować zakładników, wystarczy do nich podejść, a wtedy znikają, przenosząc się na Twój statek ratunkowy krążący po orbicie wokół Planety X. Jeśli tak się nie będzie działo, musisz zniszczyć konsolety komputerów stojących pod ścianą. Główną przeszkodę w Twej misji stanowią czarne płytki, pojawiające się na podłodze.

ESCAPE FROM THE PLANET OF THE ROBOT MONSTERS to gra labiryntowa, posiadająca niezłą grafikę i efekty dźwiękowe. Przeznaczona jest dla jednego lub dwóch graczy, którym nie obcy jest refleks.

*ROBERT KULIŚ*

**FIRMA: DOMARK**
**DYSTRYBUTOR: IPS Computer Group, Warszawa**
**RODZAJ GRY: labiryntowa, zręcznościowa**
**KOMPUTER: C-64**
**WYMAGANIA: -**

| ocena | 0 100% |
|---|---|
| grafika | ▮▮▮▮ |
| muzyka | ▯▯ |
| ogólnie | ▮▮▮▮▮ |

## PIRACY

Do pogrania w PIRACY namawiam wszystkich tych, którym znudziły się strzelanki i wszelkie symulacje. Jest to gra logiczna o świetnej grafice, cokolwiek gorszym dźwięku, w którą mogą grać dwie osoby.

Na pełnym morzu toczy się walka między piratami a żołnierzami. Statki przeciwników połączone są ze sobą siecią, która stanowi krwawe pole bitwy. Zarówno załoga piratów jak i żołnierzy liczy sobie po piętnastu osobników. Oba statki posiadają po pięć klap, przez które wypuszcza się na siatkę kolejnych wojowników. Siatka jest w niektórych miejscach poprzerywana, co stanowi jedyne utrudnienie w drodze na pokład wroga.

Poruszając się po siatce możesz wykonywać skok tylko do przodu, lecz za to w trzech kierunkach:
— w poziomie,
— na ukos do góry,
— na ukos w dół.

Zawsze przeskakujesz o jedną kratkę do przodu (w obranym kierunku) wszystkimi wojownikami Twojej załogi, jacy akurat znajdują się na siatce.

**UWAGA!** Jeśli Twój wojownik znajduje się na samym dole siatki, a Ty wyznaczysz kierunek w dół na ukos, to znajdzie się on o jedno pole dalej, ale na samej górze siatki. Podobna sytuacja jest wtedy, gdy wojownik znajdujący się na samej górze będzie musiał skoczyć na ukos do góry — wtedy znajdzie się o jedno pole dalej na samym dole.

Zabijanie wrogiego napastnika jest proste — wskocz swoim wojownikiem na miejsce, gdzie znajduje się wróg. Jeżeli wskoczysz na pokład wroga przez jedną z pięciu klap, otworzy się ona, a Ty zyskasz jednego wojownika więcej.

Celem gry jest udane natarcie na wrogi statek poprzez wskoczenie przez wszystkie pięć klap do środka i wybicie wszystkich rywali.

*ROBERT KULIŚ*

**FIRMA: DOMARK**
**DYSTRYBUTOR: IPS Computer Group, Warszawa**
**RODZAJ GRY: zręcznosciowo-logiczna**
**KOMPUTER: C-64**
**WYMAGANIA: —**

| ocena | 0 100% |
|---|---|
| grafika | ▮▮▮▮▮▮▮▮ |
| muzyka | ▯▯▯ |
| ogólnie | ▮▮▮▮▮▮▮▮▮ |

 ANKIETA

W poprzedniej części zajmowaliśmy się prawie wyłącznie osprzętem, pora więc, aby wspomnieć również o oprogramowaniu. Dzięki Waszym odpowiedziom wiemy jakie programy cieszą się Waszym uznaniem i jakimi warto się zajmować; są wśród nich programy „faworyzowane", choć w niektórych grupach trudno Wam było ustalić najlepsze produkty.

Rozpocznę od języków programowania: oczywiście dla posiadaczy C-64 najpopularniejszym dialektem (oprócz wbudowanego interpretera) jest SIMON'S BASIC zawierający szereg poleceń umożliwiających obsługę grafiki i dźwięku. Nieco mniejszym zainteresowaniem cieszy się asembler co może być spowodowane dwoma przyczynami: na rynku C-64 nadal jest bardzo wiele tanich i dobrych programów, lecz również asembler nie należy do języków najprostszych do nauki. Pozostałe języki cieszą się stosunkowo niewielkim zainteresowaniem.

Nieco inaczej wygląda sprawa Amigi. Tu zainteresowanie programowaniem jest nieco mniejsze pomimo faktu, że komputer ten daje znacznie większe pole do popisu, nie mówiąc już o dostępności szeregu znanych i cenionych kompilatorów. Zainteresowania użytkowników są tu jednak nieco równomierniej ułożone: około 16% zajmuje się programowaniem w asemblerze, 16.7% — w różnych dialektach Pascala, 17.6% nadal programuje w różnych odmianach języka BASIC. Najpopularniejszy jest jednak AMOS co pozwala mi wnioskować, że użytkownicy nadal są zainteresowani oszałamiającymi efektami (dźwięk, obraz, animacja) i stosunkowo niewielkim nakładem pracy włożonym w układanie programu.

Do najbardziej lubianych przez Was programów użytkowych należą edytory graficzne, programy muzyczne oraz... programy kopiujące. Do najmniej popularnych programów zaliczają się w kolei: programy telekomunikacyjne, programy DTP (służące do składania tekstu) oraz arkusze kalkulacyjne. Ciekawe zjawisko można zaobserwować przy bazach danych (dla Commodore 64): prawie 49% użytkowników korzysta z „innych" baz danych co w praktyce oznacza, że programy te napisali dla siebie sami użytkownicy.

Gdybym miał sporządzić na podstawie Waszych odpowiedzi listę najlepszych programów dla Amigi i C-64, to zawierałaby ona następujące tytuły:

## AMIGA:

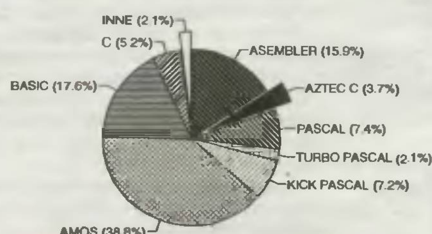
| Nazwa programu | Liczba głosów |
|---|---|
| Język programowania AMOS | 188 na 484 |
| Program graficzny DELUXE PAINT | 626 na 797 |
| Program muzyczny PROTRACKER | 172 na 683 |
| Program kopiujący X-COPY | 797 na 886 |
| Edytor tekstu CED | 307 na 497 |
| Arkusz kalkulacyjny VIP | 110 na 238 |
| Baza danych: MIAMIGAFILE 3 | 212 na 436 |
| Programy DTP PAGE STREAM | 138 na 188 |
| Narzędzia (dysk) DISK MASTER | 167 na 323 |

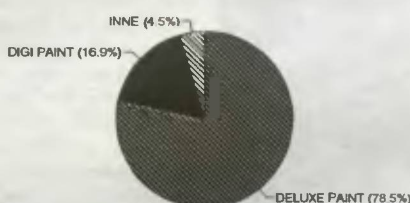
## COMMODORE 64:

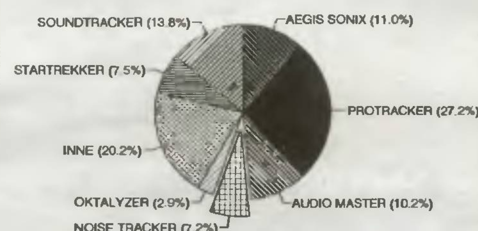
| Nazwa programu | Liczba głosów |
|---|---|
| Język programowania SIMON'S BASIC | 304 na 1031 |
| Program graficzny ADVANCED ART STUDIO | 591 na 987 |
| Program muzyczny VOICETRACKER | 286 na 603 |
| Program kopiujący FAST HACK'EM i SPECTACULAR COPY | 212 na 711<br>211 na 711 |
| Edytor tekstu POLSCRIPT | 86 na 327 |
| Arkusz kalkulacyjny: ABRACALC | 33 na 148 |
| Baza danych Inne | 126 na 281 |
| Program DTP GEOPUBLISH | 25 na 34 |
| Narzędzia (dysk) DIR MASTER | 15 na 120 |

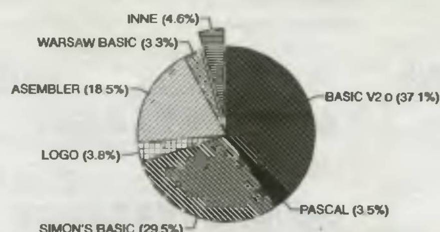
Zapraszam również Czytelników do zapoznania się z zamieszczonymi tu wykresami; są one co najmniej tak samo wymowne i jasno określają Wasze gusty i „programowe" przyzwyczajenia.

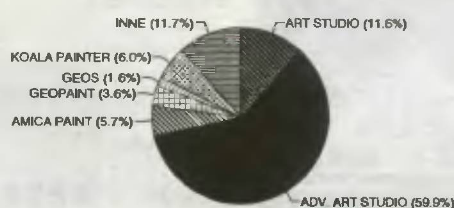
***KLAUDIUSZ DYBOWSKI***

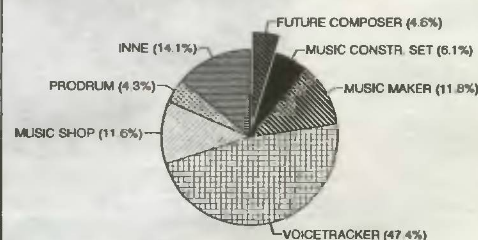


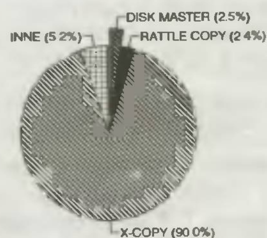



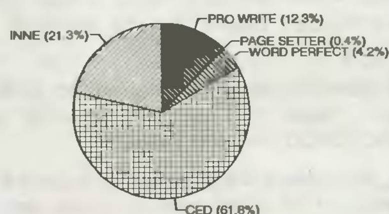



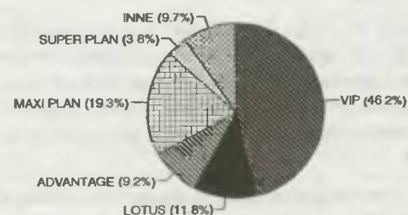
## PROGRAMY KOPIUJACE
(AMIGA - RAZEM 886 ODPOWIEDZI)



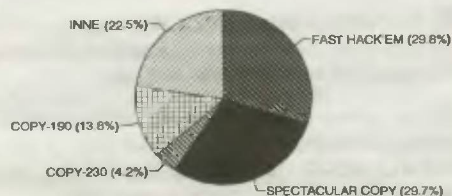

## EDYTORY TEKSTU
(AMIGA - RAZEM 497 ODPOWIEDZI)



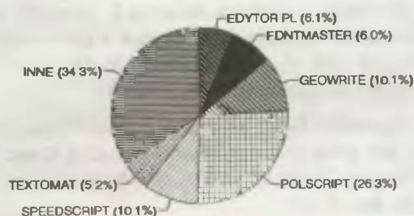

## ARKUSZE KALKULACYJNE
(AMIGA - RAZEM 238 ODPOWIEDZI)



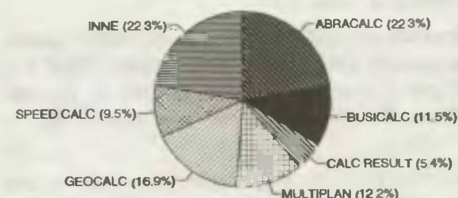

## PROGRAMY KOPIUJACE
(C-64 - RAZEM 711 ODPOWIEDZI)



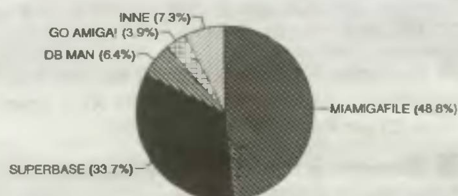

## EDYTORY TEKSTU
(C-64 - RAZEM 327 ODPOWIEDZI)



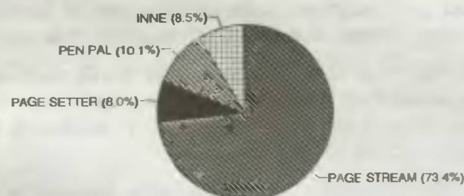

## ARKUSZE KALKULACYJNE
(C-64 - RAZEM 148 ODPOWIEDZI)



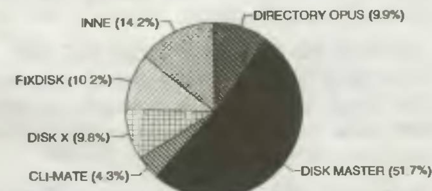

## BAZY DANYCH
(AMIGA - RAZEM 436 ODPOWIEDZI)



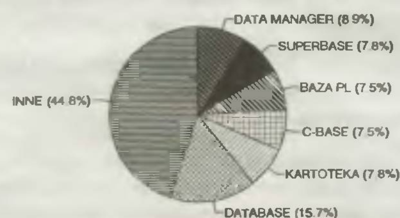

## PROGRAMY DTP
(AMIGA - RAZEM 188 ODPOWIEDZI)



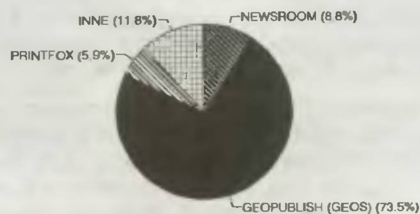

## PROGRAMY NARZEDZIOWE (DYSK)
(AMIGA - RAZEM 323 ODPOWIEDZI)



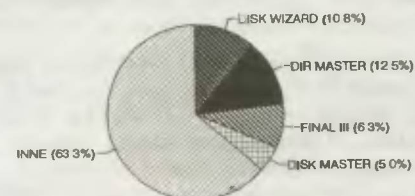

## BAZY DANYCH
(C-64 - RAZEM 281 ODPOWIEDZI)




## PROGRAMY DTP
(C-64 - RAZEM 34 ODPOWIEDZI)




## PROGRAMY NARZEDZIOWE (DYSK)
(C-64 - RAZEM 120 ODPOWIEDZI)



# SUPERMARKET

COMMODORE 64/128

■ Zamienię na Amigę 500 lub sprzedam Commodore 64VGS, stację dysków 1541II, magnetofon, joystick Megaboard, cartridge: Final III, X, Super Games, kasety i dyskietki z grami oraz literaturę. Tomasz Figzał 42 - 200 Częstochowa ul. Wodnickiego 84/90 m 16, tel. 22-13-30.

■ Sprzedam: Commodore 64, magnetofon Datenrekorder DR 1535, joystick, cartridge X, 10 kaset, literaturę. Cena 2 mln zł. Łukasz Pieczara, 91-496 Łódź, Oś. Radagoszcz, ul. Świtezianki 23 m 7.

■ Sprzedam: C-64, magnetofon DM 6403C, trzy joysticki (jeden uszkodzony), oprogramowanie, cartridge: Black Box 4.0, Tytanic. Również monitor monochromatyczny (zielona poświata) YOKO. Cena kompletu 3 mln; można się potargować! Andrzej Kieńć, 51-671 Wrocław, ul. 9 - Maja 30/1.

■ Sprzedam: C-64II (gwarancja), magnetofon (gwarancja), joystick (QS II plus), "przegrywają", Final II, X, 1200 gier i programów. Cena 2,8 mln zł. Ryszard Stępiński, 81-005 Gdynia, ul. Kcyńska 3/123, tel. 23-74-74.

■ Sprzedam: C-128, magnetofon, pióro świetlne, joysticki, literatura, oprogramowanie, moduł X. Stan urządzeń bardzo dobry. Cena około 2 mln zł. M. Pachołski, 99-100 Łęczyca, ul. Bitwy nad Bzurą 24/36, tel. 40-59.

■ Sprzedam: C-64II, dwa magnetofony, interfejs do przegrywania, joystick, cartridge Black Box i X Plus, moduł z grami, oprogramowanie i literaturę. Cena 3,2 mln zł. Piotr Kobiela, 44-335 Jastrzębie Zdrój, ul. Beskidzka 27/6.

■ Sprzedam: C-64, stacje dysków 1541II (gwarancja), magnetofon, cartridge, dwa joysticki, pudełko na dyskietki z 30 dyskami oraz 13 kaset. Cena około 4 mln zł. Kamil Wiśniewski, 82-331 Jegłownik, ul. Nogatu 15, tel. 125-07 (po godz. 15.30).

■ Sprzedam lub zamienię na Amigę 500 bądź PC AT następujący sprzęt: C-64 (gwarancja), magnetofon, joystick, Black Box, programy, motorynkę w bardzo dobrym stanie. M. Macierakowski 07-400 Ostrołęka ul. Konopnickiej 8/16.

■ Sprzedam C-64II, magnetofon, joysticki, Black Box 4, 300 programów. Cena 1,7 mln. Wojciech Chałat, 42-200 Częstochowa, ul. Aleja 8/44.

■ Sprzedam: C-128D, dyskietki, literaturę, joystick, moduł 80 znaków. Cena 4,1 mln zł. S. Samojluk, 12-100 Szczytno, ul. Marchlewskiego 10, tel. 33-52.

■ Sprzedam: C-64II (stan idealny), magnetofon DR 1535, 1541II, cartridge Final III, X, oprogramowanie i literaturę. Cena 5,3 mln zł. Paweł Krętowski, 01-390 Warszawa, ul. Puszczy Solskiej 1 m 52.

■ Zamienię (lub sprzedam) C-64C (gwar.), magnetofon, ponad 1000 programów, Black Box, Final III, literaturę, joysticki, na C-128 lub C-128D lub AMIGĘ 500 (dopłata). P. Wojewódzki, 22-400 Zamość, ul. P. Wyszyńskiego 63/29.

■ Sprzedam C-64II, magnetofon, pokrywę, 250 programów, X, Final II, joysticki. D. Koralewski, 80-506 Gdańsk, ul. Chodkiewicza 20/56.

■ Sprzedam C-128 (stan idealny), stację dysków Okeanic, Final III, magnetofon, 16 nagranych kaset, 120 dyskietek + pudełko "TAL", 2 joysticki, telewizorek prod. RFN, literaturę. Cena 4 mln. K. Jagiełowicz, 65-564 Zielona Góra, ul. Pionierów 30/2, tel. 632-89.

■ Sprzedam tanio C-64, magnetofon, joystick, cartridge i około 500 gier i programów. M. Szymczak, 63-200 Jarocin, ul. Długa 2.

■ Sprzedam C-64C (gwarancja), magnetofon, Final II, joystick TOP STAR oraz 400 gier na kasetach. Wszystko za 2,1 mln. D. Trzaska, Katowice, ul. Grabskiego 29a/19, osiedle W. Witosa, tel. 154-05-19.

■ Pilnie sprzedam C-64 VSG wraz z: magnetofonem, stacją dysków 9900, Black Box V.3, 2 joystickami, oprogramowaniem na kasetach i dyskach 5.25". R. Wieliczuk, 21-350 Międzyrzec Podlaski, ul. Warszawska 31/26.

■ Sprzedam C-64II, magnetofon, 2 joysticki, osłonę przeciwpyłową, Black Box, 5 kaset z grami i literaturę. Cena 2,3 mln zł. Michał Lutek, 26-100 Skarżysko-Kamienna, ul. Sokola 6/7, tel. 51-39-15.

■ Sprzedam C-64, magnetofon, 2 cartridge, joystick, 600 gier, programy użytkowe i literaturę. Cena 1,8 mln zł. Tomasz Olejniczak, Łódź, ul. Wyszyńskiego 106 m 27, tel. 87-76-58.

■ Zamienię C-64, manetofon, Black Box, 724 programy, wzmacniacz akustyczny, zestaw elektroakustyczny (wieża) "Diora", zestaw głośnikowy 110 W na Amigę 500. A. Jęch, 63-200 Jarocin, Oś. Konstytucji 3 Maja 23/7.

■ Sprzedam C-64II (stan idealny), magnetofon, 2 joysticki, cartridge, 300 gier, pokrywa, programy użytkowe i literaturę. Cena 1,8 mln zł. Andrzej Rosiński, 46-078 Korfantów, ul. Parkowa 7/4.

■ Sprzedam lub zamienię na Amigę 500 lub Atari 520 STE następujący sprzęt: C-64II (gwarancja), magnetofon (gwarancja), 2 joysticki, moduł X, około 1000 gier i użytków. Cena około 3 mln zł. Damian Łoziński, Gdańsk, ul. Szopy 3/40, tel. 31-95-15 w. 233.

■ Sprzedam C-64II, stację dysków 1541C, magnetofon, ponad 1000 gier na dyskach i kasetach, literatura. Cena około 6 mln. Sven Nilsson, 04-514 Warszawa, ul. Szumna 22, tel. 12-03-73.

■ Sprzedam pilnie C-64II, magnetofon, joystick, cartridge, pokrywa, literatura, 12 kaset. Cena 1,9 mln zł. Rafał Strzelecki, 97-300 Piotrków Trybunalski, ul. K. Baczyński 14 / 23g, tel. 47-02-73.

■ Sprzedam C-64II, magnetofon, cartridge, joystick, pokrywa, kasety i literatura. Cena 1,9 mln zł. Bogdan Sobierański, 90-315 Łódź, ul. Kilińskiego 141 m 13.

■ Sprzedam C-64II (gwarancja), telewizor Biazet TMP 205 (wejście monitorowe, gwarancja), Black Box, 2 joysticki, 17 kaset (z pudełkiem), wejście na dwa magnetofony, literatura - cena 3.8 mln zł. Jarosław Wita, 44-234 Leszczyny, ul. Odrodzenia 48.

■ Sprzedam C-64C, stację 1541II (11 miesięcy gwarancji), magnetofon, Final III, X, 2 moduły z testami z geografii i historii, mysz, ponad 500 programów, dyskietki pokrywę, literaturę, QuickJoy II. Razem lub osobno. Cena kompletu 5 mln zł. Radosław Drozdowski, 09-520 Łąck, ul. Słoneczna 5, tel. 140-13.

AMIGA

■ Sprzedam: Amigę 500, 1MB, SD 5,25", TVC 14", dwa joysticki, 50 dyskietek, drukarka Star LC 10 Color. Cena 13 mln. Roman Firlit, 98-291 Chałupia Mała, ul. Osmolińska 8, tel. w Sieradzu: 115-94.

■ Sprzedam modulator TV do Amigi. Cena 350 tys. zł. Michał Pawluk, 22-400 Zamość, ul. Wiejska 3/28.

■ Sprzedam: Amigę 500 z kickstartami 1.3 i 2.04, 1 MB RAM, monitor 1084 S, stacja Stodoła 5,25", filtr, sampler stereo, TOP STAR, ewentualnie pewną liczbę dysków lub zamienię na A200. Cena 13,5 mln. Kupię książkę J. Kostrzewskiego MC 68000. Tomasz Gosiewski, 21-100 Lubartów, ul. 1-go Maja 63/20, tel. (0836) 32-68.

■ Sprzedam: Amigę 500 (dwa miesiące używana) - stan idealny, kabel Euro - Scart, 13 dysków. Cena: 5,5 mln zł. Pilne! Wojciech Tomas, 76-270 Ustka, ul. Słupska 5/46.

■ Sprzedam: Amigę 500 2,3 MB, joystick, modulator TV, 200 dyskietek i literaturę. Cena 6 mln zł. Grzegorz Karolczak, 53-336 Wrocław, ul. Wielka 53/24.

■ Sprzedam: A500 1MB, monitor mono, stacja 5,25", moduł, 150 dysków, dużo dodatków. Cena 11 mln zł. Z drukarką LC 20 - 14,5 mln zł. Piotr Mikulski, 41-300 Dąbrowa Górnicza, ul. Adamieckiego 11/115.

■ Sprzedam PC AT 16 MHz, FDD 1.2, 1.44, 1 MB, Hercules, Monitor P/W, Mysz, Covox, 40 Dyskietek, gwarancja do XII 92, lub zamienię na Amigę 500, wersję angielską, Karol Sawicki, 16-400 Suwałki, ul. Kościuszki 83/7.

■ Pilnie sprzedam lub zamienię na AMIGĘ 2000 (wersja C), AMIGĘ 500, 100 dyskietek, 2 x disc box, 6x joystick, filtr z uziemieniem, literaturę i dyskietkę czyszczącą. Gwarancja do 16.04.1993. B. Daniszewski, Gdańsk 28, ul. Hallera 247a/5, tel. 43-44-24.

■ Niezawodny magnetowid SANYO + 20 filmów zamienię na AMIGĘ 500 z modulatorem. K. Krawczyk, 37-400 Nisko, ul.Reymonta 3.

■ Sprzedam AMIGE 2000, HD 40 MB, FDD 3,5 i 5,25 z monitorem kolor-stereo, karta IBM XT + programy. Cena 23 mln. Katowice, tel. 154-69-98.

■ Sprzedam w komplecie (za 17 mln zł) lub osobno: Amiga 500+ (ROM 2.0/1.3), pokrywa, roszerzenie RAM 0.5 MB, stacja dysków 3,5", Action Replay MK3, digitizer dźwięku, monitor 1084S, filtr do monitora, 160 dysków, pudełka na dyskietki, literatura. Daniel Walesiak, 11-041 Olsztyn 15, ul. Zbicza 8.

■ Sprzedam komputer Amiga 500 1 MB RAM, monitor kolorowy, rozszerzenie pamięci, joystick, 50 dyskietek z programami, filtr na monitor, podkładka pod myszkę Razem lub osobno. Jarosław Radoń, 43-300 Bielsko-Biała, ul. Wrzosowa 21, tel. 293-74 po 19.00.

■ Sprzedam Amigę 500 (1 rok, klawiatura USA). Cena 5,4 mln zł. Aleksander Petryk, 71-436 Szczecin, ul. Niemierzyńska 11/12.

■ Sprzedam A500 1 MB RAM, zegar wew., 150 dysków z oprogramowaniem, 2 joysticki, mysz, literatura, pomoc w zainstalowaniu. Wojciech Kowalski, 53-326 Wrocław, ul. Buska 4/17, tel. (0-71) 61-18-37.

OPROGRAMOWANIE

■ Wymienię oprogramowanie na C-64 (dysk). Arkadiusz Marecki, 44-240 Żory, Oś. Księcia Władysława 12f/8.

■ Wymienię lub kupię gry i programy użytkowe na C-84 (magnetofon). Tomasz Zdanewicz, 15-869 Białystok, ul. Białostoczek 11/54.

■ Wymienię się grami na C-64 (magnetofon). Piotr Szarek 32-200 Miechów, Oś. XXX-lecia 11/70.

■ Poszukuję gry Test Drive 2 na C-64 (magnetofon). A. Tocha, 83-200 Starogard Gdański, ul. Korczaka 2, tel. (0-69) 226-71.

■ Poszukuję symulatorów na C-64. W zamian oferuję wymianę lub gotówkę. Krzysiak Gawryluk, 15-814 Białystok, ul. Gan. Z. Berlinga 42/4.

■ Pilnie poszukuję programów: Font Master II, Print Fox i oprogramowania DTP na C-64. Grzegorz Baran, 35-507 Rzeszów, ul. Osmeckiego 13/9.

■ Wymienię programy demonstracyjne - DEMA. Poszukuję narzędzi do tworzenia programów demonstracyjnych. Radosław Łukawski, 26-680 Wierzbica, Łączany 7a.

■ Kupię edytory tekstu: Fontmaster II, Geowrite, PrintFox. Cezary Zadrocki, 66-400 Gorzów Wlkp., ul. Z. Berlinga 3b/16.

■ Poszukuję na C-64 (magnetofon) następujących gier: Navy Seal, Wings of Fury 2, 3, Last Ninja 1,2, North & South, interpreter Simons BASIC-a. Bartosz Dalewski, 98-355 Działoszyn, ul. A. Mickiewicza 16b.

■ Wymienię 27 kaset do C-64, cartridge Spit Ball (cartridge), grę APB, joystick PYTHON (900 tys.) na monitor zielony (bursztynowy) lub używaną stację dysków. Michał Konieczko, 81-068 Gdynia, ul. Wiejska 22/28 m 3.

■ Kupię programy na kasetach do C-64. Sprzedam Simons BASIC (170 tys.). Marcin Kaczmerek, 72-600 Świnoujście, ul. Kujawska 11/1.

■ Sprzedam oryginalne programy MIDI na C-64: Supertrack, Sequencer Scoretrack wraz z opisem oraz GEOS-a, 120 tys. każdy. Marcin Diling, 70-781 Szczecin, ul. Seledynowa 55/8.

■ Wymienię oprogramowanie, opisy i mapy na C-64 (magnetofon). Konrad Szajner, 22-400 Zamość, ul. Wąska 16/46.

■ Kupię legalną kopię programu muzycznego VOICETRACKER ver. 4.0 na C-64. Mariusz Uszyłło, 59-400 Jawor, ul. M. Konopnickiej 19.

■ Wymienię programy i doświadczenia dotyczące Amigi 500. Piotr Zieliński, 18-100 Łapy, ul. Sikorskiego 70/4.

■ Wymienię programy, gry i doświadczenia dotyczące Amigi 500. Poszukuję programów: Amigazer, Tarot (2D), Astrology (2D), Deepsky, Planets, Moonphase, Arioch's Astrology (2D).Mirosław Winczewski, 44-121 Gliwice 21, ul. Bolesława Śmiałego 4 m 1.

■ Sprzedam lub/i wymienię programy (gry, użytki) na Amigę 500/500 Plus. Poszukuję członków do swojej grupy. Jakub Pieńkowski, 12-100 Szczytno, ul. Chopina 4.

■ Wymienię gry, użytki i doświadczenia dotyczące komputera Amigi 500 (1 MB RAM). Przemysław Pochopień, 34-200 Sucha Beskidzka, Oś. XX-lecia 5/11.

INNE

■ Zamienię magnetofon 1530, cartridge, ok. 450 gier i programów użytkowych, aparat "ZENIT" z lampą błyskową na sprawną (używaną) stację dysków do C-64. Piotr Peliwoda, 14-510 Orneta, ul. 1-go Maja 48b/1.

■ Sprzedam osprzęt, oprogramowanie i literaturę do C-64/C-128. Tomasz Kępa, 87-100 Toruń, ul. Stawisińskiego 4f/68, tel. 48-85-23.

■ Sprzedam magnetofon 1530 do C-64, 20 kaset (800 tys.) oraz kupię lub wymienię następujące programy: Rings of Medusa, Fight Night, North & South, Kings of the Beach oraz Match Fishing na C-64. Rafał Siemek, 32-500 Chrzanów, ul. Orkana 21b/10, tel. (0-35) 376-62.



Odcinek dla poczty
zł ....
Słownie zł ....
Wpłacający ....
Dokładny ....
adres ....
i kod ....
Wydawnictwo BAJTEK Warszawa, ul. Wspólna 61
Bank Agrobank S.A. 470005-1834-131 ul. Grochowska 262 04-398 Warszawa
Datownik
Opłata
podpis przyjmującego

Odcinek dla posiadacza rachunku
zł ....
Słownie zł ....
Wpłacający ....
Dokładny ....
adres ....
i kod ....
Wydawnictwo BAJTEK Warszawa, ul. Wspólna 61
Bank Agrobank S.A. 470005-1834-131 ul. Grochowska 262 04-398 Warszawa
Datownik
Opłata
podpis przyjmującego

Potwierdzenie dla wpłacającego
zł ....
Słownie zł ....
Wpłacający ....
Dokładny ....
adres ....
i kod ....
Wydawnictwo BAJTEK Warszawa, ul. Wspólna 61
Bank Agrobank S.A. 470005-1834-131 ul. Grochowska 262 04-398 Warszawa
Datownik
Opłata
podpis przyjmującego

Odcinek do wysłania
zł ....
Słownie zł ....
Wpłacający ....
Dokładny ....
adres ....
i kod ....
Wydawnictwo BAJTEK Warszawa, ul. Wspólna 61
Bank Agrobank S.A. 470005-1834-131 ul. Grochowska 262 04-398 Warszawa
Datownik
odpis
podpis przyjmującego

| Tytuł \ Liczba kolejnych zeszytów | 3 | 6 | 12 | po ile egz. |
|---|---|---|---|---|
| Bajtek | X | 60000 | 120000 | |
| C&A | 30000 | 60000 | X | |
| TOP SECRET | 27000 | 54000 | X | |

Wpłat dokonywać na konto:
Spółdzielnia "BAJTEK"
Bank "Agrobank S.A."
470005-1834-131
ul. Grochowska 262
04-398 Warszawa

tu zanotuj, co zamówiłeś

## Warunki prenumeraty:

- Prenumerata zawarta przed upływem ważności kuponu gwarantuje niezmienność cen
- Przesyłka pocztowa nie wymaga dodatkowych opłat
- Jeżeli w ciągu 2 tyg. od pojawienia się numeru w kioskach przesyłka nie nadeszła, prosimy o kontakt
- Za błędy wynikające z niestarannnego wypełnienia formularza redakcja nie ponosi odpowiedzialności
- Prosimy o staranne i wyraźne zakreślenie odpowiednich ilości egzemplarzy

## NINIEJSZYM ZAMAWIAM:

| m-c | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| "C&A" | | | | | | | | | | |
| Dyskietka 5,25" dla C-64/128/16/116/+4 | | | | | | | | | | |
| Dyskietka 3,5" dla Amigi | | | | | | | | | | |

Kaseta dla C-64 ☐

IMIĘ: . . . . . . . . . . . . . . . . . . .
NAZWISKO: . . . . . . . . . . . . . . .
ADRES: . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . (Kod): . . . . . .

UWAGI:

Do kuponu dołączam kopię dowodu wpłaty.



### NAJPIERW PRZECZYTAJ POTEM ZAMAWIAJ!

1. Kupon ułatwia i przyspiesza realizację Twojego zamówienia.
2. Kupon bez potwierdzenia wpłaty jest nic nie wartym kawałkiem papieru. Zanim wyślesz do nas zamówienie upewnij się, czy włożyłeś do koperty potwierdzenie (kopię) wpłaty na nasze konto odpowiedniej sumy.
3. Wypełniaj kupon **WYRAŹNIE**, aby nie było żadnych wątpliwości, co chcesz zamówić.
4. Nie zapomnij o podaniu swojego **PEŁNEGO** adresu (miejscowość, kod pocztowy, ulica, numer domu, mieszkania).
5. Jeśli Twoje zamówienie jest nietypowe, masz miejsce w rubryce „UWAGI", aby wszystko szczegółowo wyjaśnić.

Pieniądze wpłacaj dla:
**Spółdzielnia „Bajtek", ul. Wspólna 61, Warszawa**
na konto:
**Bank Agrobank S.A.**
**470005-1834-131**
**ul. Grochowska 262**
**04-398 Warszawa**

**Ceny:**

| | |
|---|---|
| Egzemplarze archiwalne C&A | 12.000 zł/szt. |
| Dyskietka 5,25" dla C-64/128 | 21.000 zł/szt. |
| Dyskietka 3,5" dla AMIGI | 21.000 zł/szt. |
| Kaseta dla C-64 | 100.000 zł/szt. |

**UWAGA!**
Kaseta dla C-64 kosztuje sto tysięcy złotych, a nie dziesięć! Znajdują się na niej wszystkie programy publikowane w C&A nr 1-6/92.

# KĄCIK POCZĄTKUJĄCEGO II



W zeszłym miesiącu poznaliście poszczególne typy komputerów ośmiobitowych z rodziny Commodore. Dzisiaj poznacie podstawowe instrukcje języka BASIC umożliwiające komputerowi porozumiewanie się z urządzeniami zewnętrznymi (magnetofon, stacja dysków), co pozwoli Wam bez ograniczeń korzystać z programów, których na pewno już trochę zebraliście a jeszcze nie bardzo wiecie, jak je wczytać.

Zacznijmy jednak od podstaw. Po pierwsze co to jest ten BASIC? Otóż BASIC to język programowania, za pomocą którego możecie Waszemu komodorkowi wydawać polecenia do wykonania od razu (tzw. tryb natychmiastowy) lub pisać programy (tryb programowy). Komputer nie dość, że zrozumie polecenie, to jeszcze w przypadku, gdy coś źle napiszecie (np. zapomnicie jednej literki), wyświetli odpowiedni komunikat informujący, jaki błąd został popełniony. Zachowa się więc, jak istota rozumna...

**Dawniej bardzo popularny język programowania, który zapisywano na stałe w pamięci ROM komputerów (np. C-64, C-16, C-116, C-PLUS/4, C-128, ZX SPECTRUM, ATARI XE/XL i inne). Zależnie od wersji BASIC może mieć kilkadziesiąt (V2.0 w C-64) lub prawie dwieście (V7.0 w C-128) instrukcji bardzo łatwych do opanowania przez użytkownika, ponieważ są to po prostu angielskie nazwy (względnie skróty nazw) konkretnych operacji, jakie ma wykonać komputer. Np. LOAD — załaduj, READ — czytaj, IF... THEN — jeśli... to itp.).**

Na razie nie będziemy wnikali jak to się dzieje, że komputer w ogóle coś rozumie, i dlaczego rozumie angielskie słowo LOAD, a nie jego polski odpowiednik, czyli ZAŁADUJ. Przyjmijcie po prostu, że tak już jest i koniec. Również nie podlega dyskusji bezkonkurencyjność języka angielskiego w zastosowaniach komputerowych, a to ze względu na krótkie słowa i operowanie wyłącznie alfabetem łacińskim (nie ma w nim np. literek ą, ę itp., które stanowiłyby dodatkową komplikację). Nie martwcie się, jeżeli nie znacie jeszcze angielskiego. Rychło odrobicie tę zaległość, a słówka, których się nauczycie, zostaną Wam w pamięci na całe życie i przydadzą się przy pracy z każdym komputerem, nie tylko z C-64.

Druga sprawa: urządzenia zewnętrzne. Bez nich komputer mógłby konkurować co najwyżej z maszyną do pisania, ponieważ dysponowałby tylko tym, co w niego na stałe „włożył" producent. A tymczasem C-64 może przeistaczać się w maszynę matematyczną, w księgowego, w rysownika, w kompozytora itd., wystarczy tylko wczytać odpowiedni program z magnetofonu lub stacji dysków.

Tu mała dygresja: urządzeniem zewnętrznym jest także monitor (telewizor), co być może nie wszyscy z Was sobie do tej pory uświadamiali (dzisiaj monitory to rzecz oczywista, ale przecież dawniej człowiek porozumiewał się z komputerem za pośrednictwem papierowej taśmy, w której maszyna wybijała pracowicie dziurki...).

Zatem urządzenia zewnętrzne (poza monitorem) służą do przechowywania danych, które komputer traci bezpowrotnie za każdym razem, gdy zostanie wyłączony. Dawniej królowały magnetofony, obecnie są już na wymarciu — zastąpiły je doskonalsze i bardziej uniwersalne stacje dysków. W naszym kąciku będziemy jednak traktować oba te urządzenia na równych prawach.

Dość tego wstępu, zaczynamy. Masz komputer i chcesz przegrać od kolegi jakiś program. Pożyczasz więc od niego kasetę, wkładasz do magnetofonu (o stacji dysków za chwilę), a na klawiaturze komputera piszesz:

**LOAD ‹RETURN›**

Tu uwaga! ‹RETURN› oznacza, że masz **nacisnąć klawisz z napisem RETURN.** Póki go nie naciśniesz, komputer nic nie wykona. Zasada ta obowiązuje zarówno w trybie natychmiastowym, jak i w programowym, na wszystkich komputerach — naciśnięcie klawisza RETURN (na niektórych komputerach oznaczony jako ENTER) to NIEZBĘDNE zatwierdzenie wydanego polecenia.

Tak więc wydałeś swojemu C-64 polecenie LOAD (czyli ZAŁADUJ). Teraz powinien pojawić się komunikat PRESS PLAY ON TAPE (wciśnij klawisz PLAY w magnetofonie). Spełnij życzenie maszyny. Obraz zniknie i usłyszysz cichy szum pracującego magnetofonu. Gdy silnik się zatrzyma (jeśli zostanie znaleziony jakiś program), ujrzysz na ekranie dodatkowy napis:

**FOUND PROGRAM1**

Oznacza to: ZNALAZŁEM PROGRAM1. Oczywiście PROGRAM1 to tylko przykład nazwy programu (może ona być dowolna), bowiem za słowem FOUND ukaże się zawsze nazwa aktualnie odszukanego programu.

Jeśli teraz odczekasz ok. 10 sekund, komputer zacznie automatycznie wczytywać program do pamięci RAM (patrz ramka). Zamiast czekać możesz nacisnąć klawisz C= (COMMODORE LOGO) lub spację — efekt będzie ten sam.

**Z ang. Random Access Memory — pamięć o dowolnym dostępie, czyli taka, której zawartość można odczytać i/lub zmienić (czyli zapisać ją na nowo).**

**Z ang. Read-Only Memory — pamięć, z której można tylko odczytywać dane, nie można ich zmieniać.**

Aby wczytać do pamięci program zapisany na dyskietce, musisz użyć następującej instrukcji:

**LOAD "nazwa", 8 ‹RETURN›**

Pojawił się cudzysłów oraz tajemnicze wyrażenie ",8". Jeśli chodzi o to pierwsze, to system operacyjny stacji dysków (program zapisany na stałe w pamięci ROM urządzenia "wymaga", by przy instrukcji LOAD umieścić w cudzysłowie nazwę programu. Bez niej nic nie zdziałasz. Natomiast ósemka na końcu polecenia określa numer urządzenia, z którego chcesz odczytać dane. Standardowo stacja dysków oznaczona jest jako urządzenie numer 8.

Jeśli zapomnisz dodać na końcu polecenia numer urządzenia, C-64 "uzna", że chodzi o magnetofon (normalnie oznaczony jako urządzenie numer 1). W przypadku magnetofonu, jak zauważyłeś, nie trzeba też podawać nazwy programu — wczytany zostanie pierwszy napotkany na taśmie.

W poleceniach adresowanych do stacji dysków możesz używać specjalnych znaków zarezerwowanych dla jej systemu operacyjnego. Dzięki nim Twoja praca staje się znacznie prostsza. Przekonaj się o tym sam i wykonaj (nie zapomnij po każdym poleceniu nacisnąć klawisza RETURN):

**LOAD "$",8** — spowoduje wczytanie katalogu dyskietki (spis nazw programów na niej nagranych). Katalog można następnie wyświetlić poleceniem LIST ‹RETURN›.

**LOAD "*",8** — wczyta do pamięci pierwszy program z dyskietki.

**LOAD "A*",8** — wczyta do pamięci pierwszy program o nazwie zaczynającej się na literę A.

**LOAD "A?A",8** — wczyta do pamięci program o trzyliterowej nazwie, z której nie zapamiętałeś środkowej litery.

Zasada jest prosta: pytajnik zastępuje nieznany znak, natomiast gwiazdka — ciąg znaków.

**WCZYTANIE PROGRAMU DO PAMIĘCI**

***Magnetofon:***
**LOAD ‹RETURN›**
**lub LOAD "nazwa",1 ‹RETURN›**
***Stacja dysków:***
**LOAD "nazwa",8 ‹RETURN›**

Masz już więc program w pamięci komputera i chciałbyś go zgrać na swoją kasetę (lub dyskietkę). Nic prostszego! Napisz:

**SAVE "nazwa" ‹RETURN›** (magnetofon)

lub

**SAVE "nazwa",8 ‹RETURN›** (stacja dysków)

SAVE to po angielsku ZACHOWAJ. Rozkaz ten służy właśnie do zgrywania programów z pamięci komputera na nośnik magnetyczny. Pamiętaj, że przy zgrywaniu wcale nie musisz używać tej samej nazwy (może być inna niż w instrukcji LOAD) oraz, że nie wolno Ci w nazwie programu użyć znaków specjalnych: gwiazdki, przecinka, znaku zapytania, dwukropka, średnika i tzw. „małpy" (@).

**ZAPISANIE PROGRAMU NA NOŚNIKU**

***Magnetofon:***
**SAVE "nazwa" ‹RETURN›**
**lub SAVE "nazwa",1 ‹RETURN›**
***Stacja dysków:***
**SAVE "nazwa",8 ‹RETURN›**

I to już wszystko na dziś. W trakcie eksperymentów, do których Cię namawiam, na pewno pojawią się nieoczekiwane problemy, ale nie martw się — wyjaśnię je w następnym odcinku, napiszę też co nieco na temat budowy Twojego C-64, układów specjalizowanych i organizacji pamięci.

*(cdn.)*
*ARNOLD*

## DRODZY CZYTELNICY!

**Wychodząc naprzeciw Waszym oczekiwaniom poświęcamy jedną kolumnę C&A na odpowiedzi dotyczące nurtujących Was problemów. Naturalnie nie możemy — ze względu na szczupłość miejsca — odpowiedzieć na wszystkie Wasze pytania, będziemy więc wybierać te najistotniejsze i najczęściej powtarzające się.**
**Począwszy od tego numeru C&A rubryka „LISTY" będzie ukazywać się regularnie co miesiąc. Mamy nadzieję, że w ten sposób nawiążemy z Wami nieco bliższy kontakt.**
**Czekamy na Wasze opinie dotyczące rubryki oraz, oczywiście, na pytania, na które w miarę możliwości postaramy się udzielac zadowalających odpowiedzi.**

• • •

**1. Czy warto kupić mysz do C-64? Jeżeli tak, to jaka byłaby najlepsza?**
**2. Dlaczego C-64IIC jest lepszy od jego poprzednika?**
**3. Jaki jest obecnie najlepszy edytor muzyczny na C-64?**
**4. Z tyłu koło wejścia antenowego jest jakiś przełącznik. Jak się go przesunie to słychać szum. Co to jest?**
**5. Ile jest części International Karate, a ile Exploding Fist?**
**6. Czy do monitora powinno się dokupywać filtr ochrony?**
**7. Co to jest PASCAL i LOGO?**

*Łukasz*

**1.** Najpierw trzeba się zastanowić, co się przy pomocy myszy będzie robiło. Gier pracujących z myszkami na C-64 w zasadzie nie ma. Jedyny, znany mi sensowny program posiadający *driver* dla myszy 1351 to GEOS. Można także skonfigurować mysz jako joystick i używać jej w programach graficznych. Lecz warto ochłonąć z entuzjazmu i na zimno skalkulować, czy aby wydatek rzędu 150–300 tys. zł będzie warty świeczki.
**2.** Kto powiedział, że C-64IIC jest lepszy od starego C-64? Raczej jest wręcz przeciwnie. To prawda, że nowy C-64 jest, nowocześniejszy i ma ładniejszą obudowę (i wyższą cenę), lecz spora ilość wprowadzonych w nim modyfikacji wcale nie świadczy o jego przewadze nad „brzydkim" C-64. Jako przykład, rzucę hasło: układ SID i jakość samplingów, które w nowym komputerze wykonywane są prawie „językiem migowym".
**3.** Nie da się jednoznacznie określić, który z obecnie dostępnych edytorów muzycznych na C-64 jest najlepszy. Natomiast mogę wymienić kilka z nich, które przypadły mi do gustu. Są to: RO-MUZAK, FUTURE COMPOSER II, DRUMMAKER 2.0, ROCKMONITOR 5 i VOICETRACKER 4.0.
**4.** Przełącznik oznaczony literkami „L-H" (ang. Low-High) umieszczany w niektórych modelach C-64 na tylnej ściance komputera obok wyjścia TV służy do dokładniejszego dostrojenia komputera do telewizora. W zależności od rodzaju odbiornika telewizyjnego, obraz i dźwięk może być lepszy, jeżeli wysyłamy słabszy sygnał (przełącznik na Low), a czasami telewizor wymaga silnego sygnału (High).
**5.** Dotychczas na rynku ukazały się 4 części INTERNATIONAL KARATE i także 4 części EXPLODING FIST.
**6.** Tak. Ciągłe wpatrywanie się w kineskop monitora, szczególnie, gdy jest on kolorowy, bardzo męczy wzrok. W dodatku dostajemy sporą dawkę promieniowania. Obecnie światowe firmy produkują już monitory typu LOW RADIATION, lecz tylko do komputerów klasy PC. Użytkownikom innych komputerów i monitorów proponuje się zakup filtrów ekranowych. Nie dają one zupełnego zaniku promieniowania, lecz go tylko skutecznie redukują. Naprawdę, wierzcie mi, gdyby nie filtr na ekranie mojego monitora (pracuję przy komputerze ok. 9 godzin dziennie), to mógłbym się obecnie poszczycić parą dość grubych szkieł powiększających na nosie.
**7.** PASCAL i LOGO są językami programowania. LOGO uznane zostało za jeden z prostszych języków polecanych szczególnie początkującym. PASCAL jest natomiast językiem wyższego rzędu, bardziej profesjonalnym, w którym programy są pisane strukturalnie. Język ten używa się m.in. na komputerach klasy PC.

*PIOTR LISZEWSKI*

• • •

**Posiadam C-64 z magnetofonem i monitorem BIAZET. Mam kilka pytań związanych z tym komputerem:**
**1. Na monitorze mam założony filtr z siatki. Po pewnym okresie używania zakurzył się. Jak go wyczyścić?**
**2. Pragnę kupić stację dysków do mojego C-64, lecz nie wiem za bardzo, którą z dostępnych na rynku?**

*Zbyszek*

**1.** Należy siatkę namoczyć w ciepłej wodzie z dodatkiem mydła, a następnie spłukać. Siatka zrobi się pofałdowana i luźna. Trzeba wtedy pozostawić ją w suchym miejscu do wyschnięcia. Po wyschnięciu siatka ponownie się napnie.
**2.** Obecnie najpopularniejsze i najłatwiej dostępne są dwie stacje dysków: 1541 II i 9900 (Oceanic). Według mnie oba te urządzenia są dobre i zbliżone do siebie cenowo. Jeżeli chcesz dowiedzieć się czegoś więcej na ich temat, zajrzyj do wcześniejszych numerów „C&A" i „Bajtka".

*PIOTR LISZEWSKI*

• • •

**Posiadam Commodore 64 z magnetofonem. Chciałbym rozszerzyc mu pamięć do 512 KB i przyłączyć do niego stację dysków od Amigi. Czy wtedy będę mógł korzystać z gier na Amigę? Czy warto do takiego komputera jak C-64 kupić stację dysków i drukarkę? Do jakiej największej pamięci w kilobajtach można rozszerzyć pamięć C-64?**

*(nazwisko do wiadomości redakcji)*

**1.** Podłączenie do Commodore 64 napędu Amigi i rozszerzenie pamięci do 512 KB wcale nie da zgodności z Amigą. C-64 i Amiga różnią się systemem operacyjnym, strukturą wewnętrzną i wieloma innymi cechami. Tak więc są to dwa różne komputery. Programy z Amigi w jej formacie NIGDY NIE BĘDĄ działały na C-64. Ponadto podłączenie amigowskiej stacji dysków do C-64 może okazać się bardzo trudne, o ile w ogóle możliwe.
**2.** Na pewno nie jest błędem zakup drukarki i stacji dysków do C-64. Szczególnie zakup tego drugiego urządzenia w czasach rozwoju komputeryzacji, kiedy magnetofony są już przeżytkiem, jest konieczny, gdyż 90% obecnie powstających programów dla Commodore 64 działa wyłącznie ze stacją dysków.
**3.** Pamięć C-64 można rozszerzyć w zasadzie bez ograniczeń, z tym, że będzie ona wykorzystywana tylko jako ramdysk.

*PIOTR LISZEWSKI*

• • •

**(...) Katalog Kasety wchodzi bez zarzutu! Jest tylko jeden problem: katalog czyta dobrze, lista wygląda bardzo ładnie, wybieram opcję „wygraj na kasetę", OK. „Press Record and Play Tape" — też OK. Ale potem! Wgrywam to, co się nagrało, piszę RUN i... nic, zupełnie nic. Ani RUN, ani LIST! Ekran zostaje czysty, kursor mruga READY i koniec!**

*Zbyszek*

Rzeczywiście, ale program Katalog Kasety był pisany z myślą, aby ów katalog wgrywać pod edytor tekstu, np. TEXTOMAT i dalej go obrabiać i ewentualnie wydrukować na drukarce. Większość edytorów tekstowych, np. Polscript 801, 802, pracuje na zbiorach typu SEQ, a program „Katalog Kasety" nagrywa katalog w zbiorze typu PRG.
Aby skorzystać z edytorów tekstu pracujących na zbiorach typu SEQ należy nagrany zbiór PRG zamienić na SEQ. Najłatwiej zrobić to za pomocą programu przedstawionego na listingu 1. Po uruchomieniu programu odpowiadamy, czy plik typu PRG nagrany jest na kasecie, czy na dysku. Następnie podajemy nazwę źródłową pliku (tego typu PRG) i nazwę przeznaczenia. Komputer prosi o włożenie nośnika źródłowego i naciśnięcia klawisza RETURN. Po załadowaniu pliku proszeni jesteśmy o włożenie nośnika przeznaczenia i naciśnięcia klawisza RETURN. Po nagraniu pliku typu SEQ komputer pyta, czy chcemy wrócić do BASIC. Jeżeli mamy jeszcze jakiś plik do przerobienia, naciskamy klawisz „N".
Jeżeli chcemy tylko wyświetlić katalog, możemy posłużyć się programem z listingu 2.

*MARIUSZ FERDYN*

## LISTING 1

```
100 if l=1 then 230
105 print chr$ (147);"zrodlo: k-kaseta
    d-dysk"
110 get a$
115 if a$="k" then z=1:goto 130
120 if a$="d" then z=8:goto 130
125 goto 110
130 print "przeznaczenie: k-kaseta d-
    dysk"
135 get a$
140 if a$="k" then p=1:goto 155
145 if a$="d" then p=8:goto 155
150 goto 135
155 input "nazwa zrodlowa:";z$
160 input "nazwa przeznaczenia:";p$
165 print "zrodlo: ";:if z=1 then
    print "kaseta":goto 175
170 print "dysk"
175 print "przeznaczenie: ";:if p=1
    then print "kaseta":goto 185
180 print "dysk"
185 print "nazwa zrodlowa: ";z$
190 print "nazwa przeznaczenia: ";p$
195 print "w porzadku (tak/nie)?"
200 get a$:if a$="" then 200
205 if a$<>"t" then run
210 print "wloz nosnik zrodlowy i na
    cisnij return."
215 get a$:if a$<>chr$ (13) then 215
220 l=1:load z$,z,1
225 stop
230 for a=49152 to 53248
235 k=peek (a):if k<>0 then next a
240 print "wloz nosnik przeznaczenia
    i nacisnij rt."
245 get a$:if a$<>chr$ (13) then 245
250 a$=p$+",s,w"
255 open 2,p,2,a$
260 for d=49152 to a-1
265 k=peek (d):a$=chr$ (k)
270 print# 2,a$;
275 next d
280 close 2
285 print "wyjsc do basic'a (tak/nie)?"
290 get a$:if a$="" then 290
295 if a$="t" then end
300 clr:run
```

## LISTING 2

```
100     if l=1 then 145
105     print chr$ (147);"wyswietlacz katalo
        gu kasety":print:print:print "1. - k
        aseta"
110     print "2. - dysk"
115     get a$
120     if a$="1" then n=1:goto 135
125     if a$="2" then n=8:goto 135
130     goto 115
135     print:input "filename:";a$
140     l=1:load a$,n,1
145     l=0:k=49152
150     print chr$ (147)
155     a=peek (k):if a=0 then 165
160     print chr$ (a);:k=k+1:goto 155
165     poke 53280,0
170     get a$:if a$="" then poke 53280,1:
        goto 165
175     print chr$ (147):end
```

# Amiga 4000 - premiera w Kolonii

***Najnowsze wiadomości o pierwszej reprezentantce nowej serii Amig***

**W dniach 16-22 września w Kolonii w RFN odbywała się wystawa o nazwie Photokina. Na niej właśnie miała miejsce europejska premiera Amigi 4000 - komputera, na który wszyscy fani Amigi czekali od dawna.**

Gdy siedem lat temu Amiga pojawiła się na rynku, jej możliwości graficzne i muzyczne były oszałamiające. Należała też do komputerów, jak na ówczesne kryteria, bardzo szybkich. Siedem lat to jednak kawał czasu, nawet dla człowieka, nie mówiąc już nawet o komputerach. Dziś trudno już zaimponować komukolwiek możliwościami Amigi. Ani liczba kolorów, jakimi dysponuje, ani rozdzielczość obrazu, ani też ośmiobitowy dźwięk nie są już rewelacją. Wszyscy chyba zdają sobie z tego sprawę, z coraz większą też frustracją fani Amigi przyjmowali coraz to nowe możliwości konkurencji.

Wszystkie zmiany, jakie przez te wszystkie lata firma Commodore wprowadzała w sprzętowej części Amigi, miały charakter kosmetyczny i było jasne, że duży krok naprzód jest jedynym wyjściem. Aż do ostatniej chwili, czyli wystawy w Pasadenie, nie było wiadomo, czy pogłoski o nowych Amigach są prawdziwe. Na szczęście jednak potwierdziły się prawie w stu procentach - światło dzienne ujrzała Amiga 4000. Jej światowa premiera miała miejsce na wystawie "World of Commodore" w Pasadenie w USA. Pierwszy pokaz w Europie miał natomiast miejsce w Kolonii, na wystawie Photokina, na której miałem przyjemność akurat przebywać.

Zacznę od głupstw, jakie tym rezem palnęła firma Commodore konstruując nowy komputer. Pierwsze z nich to zastosowanie sterownika twardych dysków w standardzie AT-BUS, znanym z pecetów. Jest oczywiście prawdą, że napędy w tym standardzie są tańsze, jednak takie rozwiązanie ma szereg bardzo dużych wad. Dotychczas stosowany interface SCSI pozwalał na podłączenie nie tylko twardych dysków, ale i także najróżniejszych innych urządzeń, jak CD-ROM, stacje zapisywalnych dysków optycznych, streamer'y itd. AT-BUS nic takiego nie umożliwia, jest konstrukcją, od której powoli się odchodzi.

Drugim, większego chyba kalibru głupstwem, jest zastosowanie w A4000 starej kości muzycznej. W czasach, gdy nawet Atari oferuje dźwięk szesnastobitowy, takie rozwiązanie jest idiotyzmem. Pozostaje mieć tylko nadzieję, że pogłoski o kończonych już pracach nad nową kością muzyczną okażą się prawdziwe.

I to już wszystko, co tym razem zepsuła firma Commodore. Poza tym Amiga 4000 jest komputerem bardzo udanym. Zastosowano w niej procesor Motorola 68040 taktowany częstotliwością 25MHz - jest więc około trzy razy szybsza od A3000. Wreszcie, po kilku latach zwlekania, zaistalowano w Amidze prawdziwe "gęste" stacje, pozwalające na zapis od 880KB do 1.76MB na dyskietce. A4000 wyposażona jest też, podobnie jak Amiga 600, w złącze krzemowych kart pamięci standardu PCMCIA.

To wszystko nie jest jednak najważniejsze, bowiem największe zmiany dokonały się w dziedzinie grafiki. Amiga 4000 wyposażona jest w nowy zestaw procesorów specjalizowanych, pozwalających na uzyskanie fantastycznych wręcz efektów. Paletę kolorów rozbudowano z 4096 do 16.777.216, zmieniono też zasady funkcjonowania kości tak, że liczba wyświetlanych kolorów nie zależy już od rozdzielczości.

Nie dodano żadnych nowych rozdzielczości - A4000 oferuje te same, co Amigi 500+, 600 i 3000, a więc od 320x256 do 1280x512 punktów. Nie jest to jednak wadą, bowiem są one wystarczające. Nowa Amiga potrafi za to wyświetlać na ekranie jednocześnie bardzo wiele kolorów, a konkretnie 2, 4, 8, 16, 32, 64, 128, 256, 4096 lub 252.111. Obraz z A4000 był niemal nie do odróżnienia od tego, ktory generowała stojąca obok Amiga wyposażona w 24-bitową kartę OpalVision.

I to na razie tyle o Amidze 4000. Zapraszam do przeczytania obszernego reportarzu z targów Photokina, który zamieścimy w następnym, gwiazdkowym numerze C&A. Znajdziecie tam oczywiście również nowe szczegóły na temat A4000.

***Andrzej Bobek***

## Dane techniczne A4000

CPU (główny procesor): - Motorola 68040, 25MHz

PAMIĘĆ - 2MB CHIP, 2MB FAST, rozszerzalna na płycie (moduły SIMM) do 16MB

SYSTEM OPERACYJNY: - 512KB, ROM, Kickstart 3.0

ZŁĄCZA:
- klawiatury
- szeregowe (RS-232)
- monitora (RGB lub RGBI)
- stacji dysków
- krzemowych kart pamięci PCMCIA
- 2 " Mysz/Joystick
- równolegle (Centronics)
- dwóch kanałów stereo
- wewnętrzny port dla napędów AT-BUS

WEWNĘTRZNE ZŁĄCZA SYSTEMU:
- CPU (procesora) - 200-linii
- cztery złącza kart rozszerzeń w standardzie Zorro III (100 linii)
- trzy w standardzie AT
- specjalizowane złącze wizyjne

*uwaga: istnieją cztery linie złącz (po dwa typy w linii - trzy linie "Zorro III + AT" i jedna "Zorro III + video"), w praktyce oznacza to, że na raz można podłączyć tylko cztery karty*

KLAWIATURA:
- odłączalna
- 94 klawisze, wydzielony blok numeryczny - w zasadzie identyczna jak w A2000 lub A3000

MYSZ: - optyczno-mechaniczna, dwa przyciski

STACJE DYSKÓW:
- wbudowana "gęsta" stacja 3.5 cala
- pojemność od 880KB do 1.76MB
- możliwość wewnętrzego montażu dwóch dodatkowych stacji dysków: 3.5 i 5.25 cala

TWARDY DYSK: - od 120MB w górę, AT-BUS, przewidywany też SCSI

GRAFIKA:
- nowy zestaw kości specjalizowanych AGA (nazywany też AA)
- rozdzielczości od 320*256 do 1280*512 (nie licząc obszaru ramki)
- przełączalność NTSC/PAL
- 24-bitowa paleta kolorów, 16.777.216 odcieni
- 2 do 262.144 kolorów na raz na ekranie

TRYBY PRACY WYJŚĆ DLA MONITORA:
- zwykły monitor RGB lub VGA
- częstotliwość pozioma: 15-31 kHz
- częstotliwość ramki: 50-72 Hz

DŹWIĘK: - cztery 8-bitowe przetworniki, po dwa na kanał stereo

WYMIARY: - 15.25 cala gleboka, 15 cali szeroka i 5 cali wysoka



Oficjalne zdjęcie, które na wystawie otrzymałem wśród materiałów prasowych

TURBO • TURBO • TURBO • TURBO • TURBO

## Turbo Junior II




4 mikroprzełączniki
1 przycisk fire
cena: 7.75 DM x kurs zł wg. NBP

## Turbo Micro 6




6 mikroprzełączników
1 przycisk fire
ergonomiczna konstrukcja rękojeści
cena: 10.95 DM x kurs zł wg. NBP

## Turbo PRO 300




4 mikroprzełączniki
2 przyciski fire
uniwersalna rękojeść
cena: 7.25 DM x kurs zł wg. NBP

## Turbo 6, Turbo 2 super




6 mikroprzełączników,
8 mikroprzełączników
ergonomiczna konstrukcja rękojeści
precyzyjny mechanizm
duża trwałość

Turbo 6 dodatkowo:
auto fire
1 przycisk fire
cena: 13.95 DM x kurs zł wg. NBP

Turbo 2 super dodatkowo:
przełącznik auto fire/normal
2 dodatkowe przyciski fire
cena: 14.95 DM x kurs zł wg. NBP




## Turbo PROFI




8 mikroprzełączników
ergonomiczna konstrukcja
regulator prędkości
auto fire
kontrolka LED do auto fire
2 dodatkowe przyciski auto fire
cena: 15.95 DM x kurs zł wg. NBP

## Turbo PRO, Turbo PRO acryl




6 mikroprzełączników
przełącznik auto fire / normal
kontrolka LED do auto fire
2 szybkostrzelne klawisze fire
idealny do gier sportowych
cena: 15.55 DM x kurs zł wg. NBP

## Turbo IBM




analogowy joystick
do komputera IBM
precyzyjny mechanizm
wysokiej klasy potencjometry
2 przyciski fire na mikroprzełącznikach
cena: 19.95 DM x kurs zł wg. NBP

## Geos Mouse Set




mysz do C-64
z uchwytem
i mouse padem
cena: 45.00 DM x kurs zł wg. NBP




## Turbo Corder

magnetofon do C-64
doskonała jakość mechanizmu
solidna i wytrzymała budowa
12 miesięcy gwarancji
cena: 33.00 DM x kurs zł wg. NBP




## FLOPPY 9900

najnowszy model stacji dysków do Commodore 64 w pełni kompatybilna ze stacją 1541 II
30 % szybsza
mała gabarytowo i estetyczna obudowa
cicha praca
12 miesięcy gwarancji
cena: 225.00 DM x kurs zł wg. NBP

## Track Ball

To nowość zastępująca mysz
4 modele które pracują z komputerami Amiga, Atari ST i IBM
nie wymagają konserwacji
w trakcie pracy potrzebują mniej miejsca od myszki przez to są wygodniejsze w użyciu
zarówno do pracy jak i do gier
estetyczne i dokładnie wykonane
z 12 miesięczną gwarancją
cena w zależności od modelu: 69.00-89.00 DM x kurs zł wg. NBP



**W sprzedaży posiadamy także: scannery do Amigi i IBM, stacje dysków do Amigi, rozszerzenia pamięci do Amigi 500 i 500 plus - 512kB 1MB i 2MB, myszy do Amigi i IBM, osłony, pudełka i wiele innych dodatków do komputerów.**

Sklepy i hurtownie prowadzące ciągłą sprzedaż naszych towarów:

- „VADIM" Zielona Góra ul. Kupiecka 28 tel. 656-72
- „GRACOM" Malbork ul. Mickiewicza 26 tel. 33-14
- „Com-Studio" Łódź ul. Kilińskiego 42 tel. 333-680
- „METRO" Poznań ul. Ratajczaka 31 tel. 527-563
- „MICROMAN" Katowice ul. Karoliny 4 tel. 588-471 w. 226

**Inne sklepy i hurtownie zapraszamy do współpracy.**

**„PROABIT"**
**Raszyn k/Warszawy**
**ul. Mickiewicza 14**
**kod 05-090 tel./fax 56-08-91**

**HURT • DETAL • WYSYŁKA**

TURBO • TURBO • TURBO • TURBO • TURBO